EXMARaLDA -

ein System zur Diskurstranskription auf dem Computer

Thomas Schmidt (thomas.schmidt@uni-hamburg.de)

Sonderforschungsbereich 538
Mehrsprachigkeit
Max Brauer-Allee 60
D-22765 Hamburg

Abstract

EXMARaLDA is a system for computer transcription of spoken discourse that is being developed at the SFB ‚Mehrsprachigkeit' as a basis of a multilingual discourse database into which the transcriptions in use at the SFB will be integrated at a later point in time.
The present paper describes the theoretical background of the development – a formal model of discourse transcription based on the annotation graph formalism (Bird/Liberman (2001)) – and its practical realisation in the form of an XML-based data format and several tools for input, output and manipulation of the data.

Einleitung

Die vorliegende Arbeit beschreibt EXMARaLDA, ein System zur Diskurstranskription auf dem Computer, das am SFB „Mehrsprachigkeit" entwickelt wird. Die Notwendigkeit für ein solches System ergab sich aus dem Vorhaben, die zahlreichen am SFB vorhandenen und noch zu erstellenden Transkriptionen gesprochener Sprache, die die Grundlage der Arbeit in vielen Teilprojekten bilden, in einer gemeinsamen mehrsprachigen Diskurs-Datenbank zu bündeln. Es stellte sich dabei heraus, dass die jeweils verwendeten Transkriptionssysteme[1] sowohl technisch als auch konzeptuell untereinander nicht kompatibel sind, und ein Datenaustausch bzw. eine Datenintegration in eine gemeinsame Oberfläche somit schwer oder sogar unmöglich ist. Wie das folgende Zitat zeigt,

> „While the utility of existing tools, formats and databases is unquestionable, their sheer variety – and the lack of standards able to mediate among them – is becoming a critical problem. Particular bodies of data are created with particular needs in mind, using formats and tools tailored to those needs, based on the resources and practices of the community involved. Once created, a linguistic database may subsequently be used for a variety of unforeseen purposes, both inside and outside the community that created it. Adapting existing software for creation, update, indexing, search and display of ‚foreign' databases typically requires extensive re-engineering. Working across a set of databases requires repeated adaptations of this kind."[2]

ist dies ein keineswegs ungewöhnliches Problem. Die Notwendigkeit, Diskurstranskriptionen in einer allgemein und flexibel nutzbaren Form auf dem Computer zu speichern und die Mög-

[1] Es sind dies vor allem HIAT-DOS (Ehlich (1992 und 1993)), LAPSUS (Crysman (1995)) und syncWriter (Rehbein et al. (1993))
[2] Bird/Liberman(2001: 2)

lichkeiten, die sich daraus für die sprachwissenschaftliche Praxis ergeben können, wurden erst in den letzten Jahren deutlich.
Mittlerweile existieren einige Arbeiten, die vielversprechende Ansätze für eine solche Standardisierung von computerkodierten Diskurstranskriptionen bieten, und auf denen EXMARaLDA aufbaut.

Das erste Kapitel dieser Arbeit stellt dar, welche theoretischen Überlegungen in die Entwicklung von EXMARaLDA eingeflossen sind. Im ersten Abschnitt wird zunächst das Grundprinzip der Trennung von Inhalt und Darstellung von Transkriptionsdaten dargelegt. Die folgenden Abschnitte 2 und 3 beschreiben dann die Grundelemente, aus denen sich Inhalt bzw. Darstellung herkömmlicher Diskurstranskriptionen zusammensetzen. In den Abschnitten 4 und 5 schließlich wird gezeigt, wie diese Grundelemente erweitert werden müssen, um auch komplexer strukturierte Transkriptionsdaten umfassen und deren fortgeschrittene maschinelle Verarbeitung ermöglichen zu können.
Das zweite Kapitel zeigt, wie diese theoretischen Überlegungen bei der Implementierung von EXMARaLDA in die Praxis umgesetzt werden. Im ersten Abschnitt werden Technologien vorgestellt, die geeignet erscheinen, die angestrebte technische Unabhängigkeit und Flexibilität des Systems zu gewährleisten. Der zweite Abschnitt legt kurz dar, wie diese Technologien bei der Realisierung der im ersten Abschnitt angestellten Überlegungen zum Einsatz kommen.

Das Thema der Diskurstranskription auf dem Computer lässt sich aus vielen Blickwinkeln betrachten, die teilweise rein sprachwissenschaftlicher (z.B. text-linguistischer oder diskurstheoretischer) Natur sein können (vgl. hierzu auch Ochs (1979)), teilweise auf informatische Aspekte (z.B. Software-Ergonomie) beschränkt sein können. Es ist nicht das Ziel dieser Arbeit, all diese Aspekte abzudecken. Vielmehr soll die hier gewählte Perspektive, die sich gut mit dem Terminus der „Texttechnologie“ (vgl. hierzu Lobin (1999a)) beschreiben lässt, als mögliche Schnittstelle zwischen den beiden Disziplinen aufgefasst werden. Dies geschieht nicht zuletzt in der Hoffnung, einen Ansatzpunkt zu finden, an dem sich die Nutzer und die Entwickler eines Systems zur Diskurstranskription auf dem Computer „auf halbem Wege“ treffen können.

Teil A: Theoretische Aspekte – Ein formales Modell zur Beschreibung von Diskurstranskriptionen

1. Inhalt vs. Darstellung

Die Praxis der Diskurstranskription ist älter als der verbreitete Einsatz von Computern in den Sprachwissenschaften. Ursprünglich wurden hand- oder maschinenschriftliche Diskurstranskriptionen hauptsächlich zu dem Zweck angefertigt, den flüchtigen auditiven und/oder visuellen Eindruck, den die Aufnahme eines Diskurses hinterlässt, schriftlich zu fixieren, um dem Forscher so eine detaillierte Analyse zu ermöglichen.
Der Einsatz des Computers zur Diskurstranskription war zunächst auch für diesen Zweck hilfreich – die Erstellung von Transkripten wurde vereinfacht, weil Korrekturvorgänge am Bildschirm leichter zu bewerkstelligen sind als auf dem Papier, und die einfache und kostengünstige Möglichkeit der digitalen Kopie eröffnete neue Möglichkeiten des Datenaustausches.[3]
Gleichzeitig ergaben sich durch den Einsatz des Computers aber auch neue Möglichkeiten der Datenanalyse. Statistische Auswertungen und systematisches Durchsuchen von Transkriptionsdaten konnten automatisiert werden, wodurch es möglich wurde, wesentlich größere Datenmengen zu analysieren und qualitative Ergebnisse quantitativ zu belegen.
Ein System zur Diskurstranskription auf dem Computer fungiert also als Grundlage sowohl einer menschlichen als auch einer maschinellen Verarbeitung von Transkriptionsdaten. Mit den Worten von MacWhinney (1995):

> „Slobin's view of the pressures shaping human language can be extended to analyze the pressures shaping a transcription system. In many regards, a transcription system is much like any human language. It needs to be clear in its marking of categories, and still preserve readability and ease of transcription. However, unlike a human language, a transcription system needs to address two different audiences. One audience is the human audience of transcribers, analysts and readers. The other audience is the digital computer and its programs."[4]

Die hier erwähnten Anforderungen an ein Transkriptionssystem, die sich unter den Termini ‚Klarheit' („clear in its marking of categories") und ‚Einfachheit' („readability and ease of transcription") zusammenfassen lassen, konkurrieren aber nicht nur untereinander, sondern definieren sich auch unterschiedlich, je nachdem ob der Mensch oder der Computer als Adressat des Systems angenommen wird: während der Computer zwar problemlos mit großen Datenmengen und komplexen (d.h. z.B. tief verschachtelten) Datenstrukturen umgehen kann, ist er auf eine explizite Kodierung aller relevanten Information angewiesen. Dem Menschen hingegen sind hinsichtlich Datenmenge und Komplexität von Datenstrukturen enge Grenzen gesetzt, er ist dafür aber in der Lage, implizit vorhandene Information wahrzunehmen. Vereinfachend lässt sich sagen, dass eine computerorientierte Datenrepräsentation zugunsten der Klarheit Abstriche bezüglich der Einfachheit machen kann (und muss), während es sich bei einer am menschlichen Benutzer orientierten Datenrepräsentation umgekehrt verhält. Ein Beispiel mag dies verdeutlichen:

---

[3] Siehe hierzu auch die Einleitung zu MacWhinney (1995), wo fünf historische Phasen der Analyse von Spracherwerbsdaten unterschieden werden, die der Autor mit den Schlagworten „impressionistic observation", „baby biographies", „transcripts", „computers" und „connectivity" überschreibt. Die Ausführungen in diesem Abschnitt betreffen nur die letzten drei dieser Phasen.
[4] MacWhinney (1995: 14)

```
MAX: Das sag ich doch.
      PRO   V   PRO  PART
TOM:             Ach so.
```

Für den menschlichen Betrachter dieses Ausschnittes aus einer Diskurstranskription, der so oder ähnlich auch in einer (einfachen ASCII-Text-) Datei auf dem Computer kodiert sein könnte, erschließen sich aufgrund gewisser Lesegewohnheiten und -erwartungen neben den explizit vorhandenen Zeichenketten auch implizit in diesen enthaltene Informationen. Zum Beispiel erkennt er, dass die Symbole der zweiten Zeile Annotationen zu den räumlich über ihnen stehenden Einheiten sind. Er erkennt weiterhin, dass es sich bei diesen Einheiten um Wörter handelt, dass diese zu Äußerungen zusammengefasst sind, und dass zu jeder Äußerung der zugehörige Sprecher am Anfang der Zeile vermerkt ist. Aus der räumlichen Anordnung der beiden Äußerungen zueinander ist schließlich ersichtlich, dass und wie diese sich teilweise überlappen.
Für eine computergestützte Verarbeitung dieses Transkriptionsausschnittes ist es hingegen notwendig, diese impliziten Informationen explizit zu kodieren. Eine diesbezüglich optimierte Kodierung könnte z.B. so aussehen[5]:

```
<utterance who="MAX"><word POS="PRO">Das</word> <word POS="V">sag</word>
<word POS="PRO">ich</word> <overlap nr=1><word POS="PART">doch</word>.
</overlap></utterance>
<utterance who="TOM"><overlap nr=1><word>Ach</word></overlap>
<word>so</word>.</utterance>
```

Es dürfte kaum Zweifel daran geben, dass eine solche Repräsentation für den menschlichen Benutzer schwer bis gar nicht lesbar ist. Sie ist zwar insofern klarer, als sie die impliziten Informationen expliziert, dies geht jedoch auf Kosten ihrer Einfachheit und damit der Verarbeitbarkeit für den menschlichen Benutzer.

Beim Entwurf eines Systems zur Diskurstranskription auf dem Computer müssen diese beiden gegensätzlichen Gewichtungen - Einfachheit über Klarheit für den menschlichen Benutzer, umgekehrt für die maschinelle Verarbeitung - berücksichtigt werden. Da zweifelsfrei beide relevant sind, bieten sich zwei Möglichkeiten an:

> **1.** Für die Datenrepräsentation wird ein *Kompromiss* zwischen am menschlichen Benutzer orientierter Repräsentation und am Computer orientierter Repräsentation gewählt.
>
> **2.** Das System *trennt* die beiden Ebenen und stellt Mechanismen zur Verfügung, mit denen aus einer computer-optimierten Datenrepräsentation eine für den menschlichen Benutzer optimierte Datenrepräsentation gewonnen werden kann (oder umgekehrt).

Für das hier beschriebene System wurde die zweite Möglichkeit gewählt: die interne Datenrepräsentation des Systems ist für die maschinelle Verarbeitung der Daten optimiert (eine solche Repräsentation soll ab hier der Einfachheit halber ‚Inhalt' genannt werden), und das System stellt Möglichkeiten zur Verfügung, aus dieser (automatisch) eine für den menschlichen

[5] Diese Kodierung ist grob an die Vorschläge der Text Encoding Initiative (Burnard (1995)) zur Kodierung von Transkriptionen gesprochener Sprache angelehnt.

Benutzer optimierte Datenrepräsentation (ab hier: ‚Darstellung'[6]) zu gewinnen. Dass nicht der umgekehrte Weg – d.h. die Berechnung des Inhaltes aus seiner Darstellung – gewählt wurde, ist zum einen alleine deshalb naheliegend, weil ja der Ausgangspunkt der Berechnung in jedem Falle, der Zielpunkt jedoch nicht unbedingt vom Computer interpretiert werden muss. Die Vorgehensweise, den Inhalt und nicht die Darstellung einer Diskurstranskription zu kodieren, hat jedoch noch einen weiteren Vorteil, der in folgendem Zitat zum Ausdruck kommt:

> "There is an important distinction in the encoding of electronic texts between underlying representation [i.e. Inhalt, T.S.] and display [i.e. Darstellung, T.S.]. The TEI approach focuses on an underlying representation, while acknowledging that this can be transformed to a variety of formats for particular processing purposes or for display.
> In focusing on an underlying representation it is possible to reduce a great deal of the variety in the transcription of speech. Transcribers have been very much concerned with finding a visual display which eases the (manual) processing of the material. But there is far more variation in display than in the features which the display is intended to represent."[7]

Zum einen kann also durch die Entscheidung, statt der Darstellung den Inhalt einer Transkription zu kodieren, die Varianz, die sich in verschiedenen Systemen zur Diskurstranskription findet, reduziert und so verschiedene Systeme auf eine gemeinsame Basis gestellt werden. Zum anderen bleibt das System durch die Trennung von Inhalt und Darstellung für verschiedene, an die jeweiligen Bedürfnisse des menschlichen Benutzers angepasste Darstellungsmöglichkeiten offen.
Die Diskussion zwischen Edwards (1992) und MacWhinney/Snow (1992) befasst sich unter anderem ebenfalls mit diesem Aspekt. MacWhinney/Snow bemerken dazu:

> „In the end, disputes about the exact shape of transcript displays are much like disputes over the quality of wines. A gracious host may provide a variety of different bottles in the hope of pleasing each guest. With transcript formats we can do even better – we can write a program to allow each researcher to see the data in a different way."[8]

Es scheint also, als ob die Trennung zwischen Inhalt und Darstellung allgemein als sinnvolle und notwendige Voraussetzung für ein System zur Diskurstranskription auf dem Computer akzeptiert ist. Es lässt sich jedoch feststellen, dass dabei unterschiedliche Prioritäten gesetzt werden. So argumentiert Knowles (1995):

> "The new opportunities [i.e. computer technology, T.S.] are not yet being fully recognized and exploited by linguists [...] Texts are still seen as objects in book format, with words running in horizontal lines from left to right. Annotations are added to these horizontal lines. But book format is an attribute not of speech, but of Western writing systems. There is no reason beyond established custom and practice to present speech in this way. On the contrary, since there are often several annotations relating to the same piece of data, book format is in many cases inappropriate. The use of book format without consideration of other possibilities is based on a confusion between the organization of the data itself, and the presentation of the data on the printed page."[9]

Die computer-optimierte Kodierung des Inhaltes wäre demnach nicht nur vorrangig, es wird

[6] Die Dichotomie „Inhalt vs. Darstellung" findet sich an anderer Stelle z.B. unter den Begriffen „logische vs. graphische Struktur" (Wohlberg (1999)) , „ content-based vs. form-based markup" (in der Terminolgie der Dokument-Technologie), „inhaltliche Strukturierung" vs. „Form" (Lobin 1999b), „underlying representation" vs. "(visual) display" (Johansson 1995) oder "organization of the data" vs. " presentation of the data" (Knowles 1995)
[7] Johansson (1995: 82)
[8] MacWhinney/Snow (1992: 463)
[9] Knowles (1995: 208)

sogar bezweifelt, dass es immer eine für den menschlichen Benutzer geeignete Darstellung (denn eine solche ist das erwähnte „book-format“) gibt, die alle in einer Diskurstranskription zu kodierenden Informationen berücksichtigt. Sinclair (1995) hingegen argumentiert:

> „Tamás Varádi [vom TELRI-Projekt, T.S.] said quite clearly that a code which is convenient for a machine, which may be essential for a machine, may be extremely awkward for humans. For these new conventions [hier: TEI, T.S.] to gain acceptance in the research community, there will have to be software interpreters provided, that will allow us to move between human and machine without any expenditure of effort whatever. I can't see any alternative to this, and I doubt if anyone who is managing a large spoken resource would want to dispute it. Humans who are going to use transcribed text will want human conventions as much as the transcribers. They will want to see in front of them something that is reasonably readable."[10]

Die Notwendigkeit einer für den Computer optimierten Datenrepräsentation wird hier zwar nicht abgestritten, es wird jedoch der für den menschlichen Benutzer optimierten Darstellung insofern Vorrang eingeräumt, als diese für eine Akzeptanz in der Forschergemeinschaft – und damit bei den potentiellen Nutzern eines Systems – von größerer, letztendlich ausschlaggebender, Bedeutung ist.
Da zweifellos beide Argumente ihre Berechtigung haben, wird im hier beschriebenen Transkriptionssystem versucht, weder der Inhalts- noch der Darstellungskodierung einer Diskurstranskription Vorrang einzuräumen, sondern im Gegenteil – im Sinne des obigen Zitats – ausdrücklich Wert darauf gelegt, dass das System neben einer computer-geeigneten Inhaltsrepräsentation der Daten auch Komponenten zu deren angemessener Darstellung für den menschlichen Benutzer beinhaltet. In den nächsten beiden Abschnitten wird dargestellt werden, nach welchen Prinzipien dabei vorgegangen werden kann.

2. Elemente einer Diskurstranskription

Gemäß der oben beschriebenen Trennung von Inhalt und Darstellung wird in diesem Abschnitt die Frage behandelt, welche Elemente für die inhaltsseitige Kodierung einer Diskurstranskription relevant sind, d.h. es wird über verschiedene konkrete Darstellunstypen (die dann im nächsten Abschnitt beschrieben werden) abstrahiert und versucht, diejenigen Elemente zu identifizieren, die die Inhaltsseite einer Diskurstranskription ausmachen. Als Ausgangspunkt soll dabei folgendes Zitat dienen:

> „The transcription presents who said what, in which order, events that might have happened during the spoken dialogue etc.“[11]

Demnach sind zumindest die folgenden Elemente Bestandteile einer Diskurstranskription:

1. eine Menge von *Sprechern* („who“)
2. eine Menge von verbalen („said what“) und nicht-verbalen („events that might have happened“) *Ereignissen*, bzw. eine *Beschreibung* solcher Ereignisse. Zumindest die verbalen werden dabei i.d.R. einem Sprecher zugeordnet („who said what“).
3. eine *zeitliche Ordnung* dieser Ereignisse („in which order“)

[10] Sinclair (1995: 107)
[11] Dybkjær et al. (1998b: 4)

Die Unterscheidung zwischen verbalen und nicht-verbalen Ereignissen deutet bereits an, dass in der Regel ein viertes Element hinzukommt. Bird/Liberman sagen:

> „All annotations of recorded linguistic signals require one unavoidable basic action: to associate a label, or an ordered set of labels, with a stretch of time in the recording(s). Such annotations typically distinguish labels of different types, such as spoken words vs. non-speech noises“[12]

Zusätzlich erfolgt also meist

4. eine *Kategorisierung* („different types“) der Ereignisse.

Demnach besteht das Transkribieren als Prozess also darin, einen Diskurs (bzw. die Aufnahme eines Diskurses) in eine Anzahl von zeitlich geordneten Ereignissen aufzuteilen und diesen Ereignissen jeweils eine Beschreibung, einen Sprecher und eine Kategorie zuzuordnen; und die Transkription als Datum besteht in einer geeigneten Beschreibung dieser Zuordnung.

Es kann angenommen werden, dass das Vorhandensein dieser Elemente einen Diskurs bereits für viele Zwecke *ausreichend* charakterisiert, d.h. die in der beschriebenen Zuordnung enthaltenen Informationen erlauben es bereits – wie es der primäre Zweck einer Transkription ist – den zeitlichen Ablauf eines Diskurses nachvollziehbar zu beschreiben. In den nächsten Abschnitten wird gezeigt werden, dass einerseits diese Informationen auch die mindest *notwendigen* für eine angemessene Diskursbeschreibung sind, d.h. dass weniger Information nicht für eine Transkription im herkömmlichen Sinne ausreichen würde, und dass andererseits für gewisse Darstellungsformen (vor allem die meisten vertikalen Darstellungsformen) und viele Bearbeitungsmethoden (vor allem die Annotation) *zusätzliche* Informationen nötig sind.

Es kann aber zunächst festgehalten werden:

Eine **Basistranskription** B ist eine Zuordnung

B: E → D x T x T x S x C, wobei

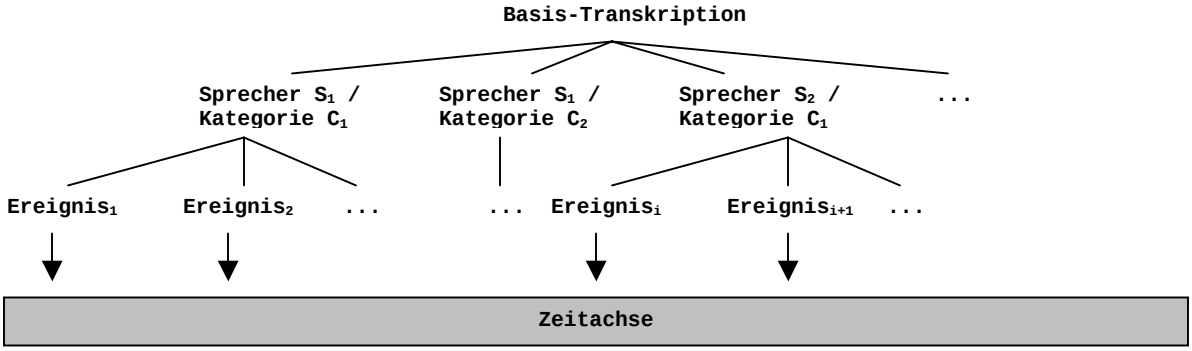
E eine (endliche) Menge von Ereignissen (der Diskurs)
D eine (endliche) Menge von Ereignisbeschreibungen
T eine Zeitachse, d.h. eine (endliche) Menge von Zeitpunkten mit einer Ordnungsrelation
S eine (endliche) Menge von Sprechern und
C eine (endliche) Menge von Ereigniskategorien ist.

Fast alle Transkriptionssysteme organisieren diese Zuordnung noch zusätzlich in Strukturelementen, die sie ‚Spur'[13], ‚Tier'[14], ‚Layer'[15] etc. nennen, und in denen Ereignisse jeweils eines

[12] Bird / Liberman (2001: 18)
[13] z.B. HIAT, vgl. Ehlich/Rehbein (1976)
[14] z.B. CHAT, vgl. MacWhinney (1995)
[15] vgl. Bird/Liberman (2001: 13)

> „[...] annotations are often stratified, where each layer describes a different property of a signal.“

Sprechers und einer Kategorie zusammengefasst sind. In der Regel gelten für die sich so ergebenden Teilmengen von Ereignissen zusätzliche Beschränkungen, wie z.B., dass sich Ereignisse in einer Spur zeitlich nicht überschneiden dürfen (dies kann aus darstellungstechnischen Gründen sogar notwendig sein – siehe Abschnitt 3) oder dass sie zuimndest in eine hierarchische Ordnung zu bringen sind (siehe dazu auch Abschnitt 4)[16]:



Wie und welche Ereignisse ausgewählt werden und wie ihre Zuordnung zu Ereignisbeschreibungen und -kategorien erfolgt, ist abhängig von der Zielsetzung der Transkription und wird in der Regel in Transkriptionskonventionen festgelegt. Die zeitliche Einordnung der Ereignisse und ihre Zuordnung zu einem Sprecher sollten hingegen unabhängig von der Zielsetzung der Transkription erfolgen können, denn sie sind mehr oder weniger objektiv feststellbar.

Das folgende Beispiel zeigt, wie die inhaltliche Repräsentation zu einer gegebenen Diskurstranskription aussehen kann:

Darstellung (Partitur)

```
MAX:   Du fällst mir immer ins Wort.              Siehst Du, Du hast es schon wieder getan.
       --fuchtelt mit den Armen ----
TOM:                          Stimmt ja gar nicht.
                              --abfällige Geste--
```

Inhalt (Basis-Transkription)

$E = \{e_1, e_2, e_3, e_4, e_5, e_6, e_7\}$
$T = \{t_0, t_1, t_2, t_3, t_4\}$ $\quad t_0 < t_1 < t_2 < t_3 < t_4$
D = {“Du fällst mir immer ins“, „Wort.“, ..., „abfällige Handbewegung“}
S = {MAX,TOM}
C = {verbal, non-verbal}

Diskurs	Basis-Transkription B(e)				
E	**D**	**T(Start)**	**T(Ende)**	**S**	**C**
e_1	Du fällst mir immer ins	t_0	t_1	MAX	verbal
e_2	Wort.	t_1	t_2	MAX	verbal
e_3	Siehst Du, Du hast es schon wieder getan.	t_3	t_4	MAX	verbal
e_4	fuchtelt mit den Armen	t_0	t_2	MAX	non-verbal
e_5	Stimmt	t_1	t_2	TOM	verbal
e_6	ja gar nicht.	t_2	t_3	TOM	verbal
e_7	abfällige Handbewegung	t_1	t_3	TOM	non-verbal

[16] Manche dieser Beschränkungen reflektieren die Realität eines Diskurses, z.B. können sich zwei sprachliche Ereignisse desselben Sprechers, die mit orthographischer Transkription beschrieben werden, ohnehin nicht überlappen, da kein Sprecher zwei solche sprachliche Handlungen zur gleichen Zeit vollziehen kann. Andere reflektieren jedoch lediglich darstellungstechnische Zwänge.

Der hier schon angedeutete Zusammenhang zwischen Basis-Transkription und Partitur-Darstellung wird im nächsten Abschnitt detaillierter erläutert werden.

3. Darstellungstypen für Diskurstranskriptionen

Edwards (1993) gibt einen Überblick über verschiedene Transkriptionssysteme. Unter dem Stichwort „spatial arrangement“ unterscheidet sie dabei drei Typen der räumlichen – d.h. darstellungsseitigen – Organisation von Transkriptionen:

> **1.** vertikale Darstellung : Die Ereignisbeschreibungen werden in *Zeilen* organisiert. Ereignisbeschreibungen innerhalb einer Zeile sind dem selben Sprecher zugeordnet. Die Reihenfolge der Zeilen spiegelt die zeitliche Abfolge des Diskurses wider.

```
MAX: [fuchtelt mit den Armen] Du fällst mir immer ins <Wort>.
TOM: [abfällige Geste] <Stimmt> ja gar nicht.
MAX: Siehst Du, Du hast es schon wieder getan.
```

> **2.** Spaltendarstellung : Die Ereignisbeschreibungen werden in *Spalten* organisiert. Jede Spalte nimmt Ereignisbeschreibungen eines Sprechers und einer Ereigniskategorie auf. Der zeitliche Ablauf des Diskurses wird in der vertikalen Anordnung der Elemente zueinander widergespiegelt (je weiter unten eine Ereignisbeschreibung steht, desto später tritt das zugehörige Ereignis im Diskurs auf, Ereignisbeschreibungen auf gleicher Höhe beschreiben simultane Ereignisse).

```
MAX                                                TOM
Du fällst mir immer ins                  fuchtelt  mit
Wort.                                    den Armen     Stimmt              abfällige Geste
                                                       ja gar nicht.
Siehst Du, Du hast es schon wieder getan.
```

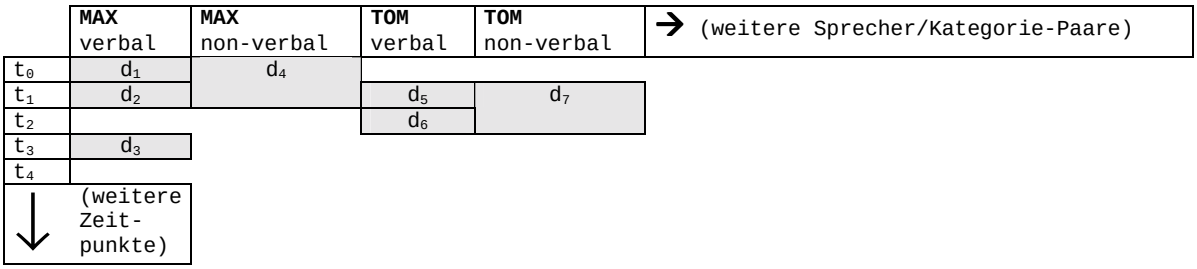
> **3.** Partiturdarstellung : Die Ereignisbeschreibungen werden in *Partiturzeilen* organisiert. Jede Partiturzeile nimmt Ereignisbeschreibungen eines Sprechers und einer Ereigniskategorie auf. Der zeitliche Ablauf des Diskurses wird in der horizontalen Anordnung der Elemente zueinander widergespiegelt (je weiter rechts eine Ereignisbeschreibung steht, desto später tritt das zugehörige Ereignis im Diskurs auf, untereinander stehende Ereignisbeschreibungen beschreiben simultane Ereignisse).

```
MAX:    Du fällst mir immer ins Wort.                  Siehst Du, Du hast es schon wieder ge-
tan.
        --fuchtelt mit den Armen ----
TOM:                             Stimmt ja gar nicht.
                                 --abfällige Geste--
```

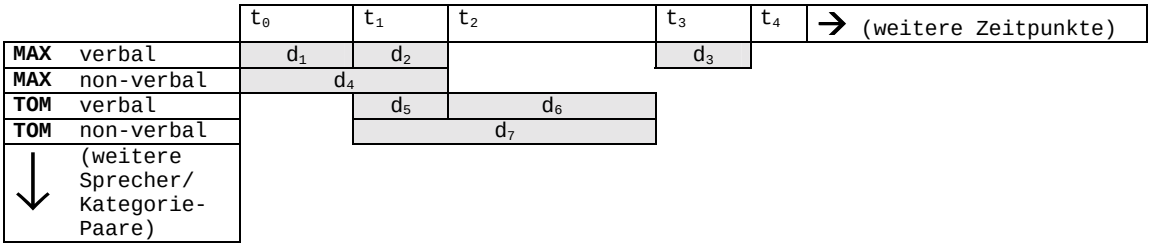
Auch wenn sich verschiedene Transkriptionssysteme in den Details ihrer konkreten Darstellung stark unterscheiden, so kann doch davon ausgegangen werden, dass sie sich (sofern sie überhaupt eine darstellungsseitige Datenrepräsentation berücksichtigen) alle einem dieser drei

Typen zuordnen lassen[17].
Spalten- und Partiturdarstellung beruhen dabei eigentlich auf dem gleichen Darstellungsprinzip; sie unterscheiden sich nur durch den Rollentausch von vertikaler und horizontaler Achse. Es wurde im vorigen Abschnitt schon angedeutet, dass sich diese beiden Darstellungstypen aus den in einer Basis-Transkription vorhandenen Informationen berechnen lassen: Die Darstellung erfolgt dabei in einem *zweidimensionalen* Koordinatensystem. Eine Achse dieses Systems entspricht der Zeitachse `T`, auf der anderen Achse werden Paare `(s,c)` von Sprechern und Ereigniskategorien abgetragen[18]. Für eine Zuordnung $B(e) = (d, t_s, t_e, s, c)$ eines Ereignisses e zu einer Beschreibung d, einem Start- und Endpunkt t_s und t_e, einem Sprecher s und einer Ereigniskategorie s wird die Ereignisbeschreibung d zwischen den Start- und Endpunkt auf der Zeitachse und zu dem ihr entsprechenden Sprecher-/Ereigniskategorie-Paar auf der jeweils anderen Achse positioniert[19]:

Spalten-Darstellung



Partitur-Darstellung:



Die vertikale Darstellung ist hingegen *eindimensional*. Sie basiert auf der Annahme, dass Diskurse *in erster Linie* sequentielle Struktur haben (d.h. etwa „ein Sprecher zur selben Zeit“[20]). So können auf der vertikalen Achse Tripel von Sprechern und Start-/Endpunkten abgetragen werden und diesen die entsprechenden Ereignisbeschreibungen zugeordnet werden. Für Dis-

[17] Rein exemplarisch seien hier CHAT (MacWhinney(1995)) und GAT (Selting et al. (1998)) als Beispiele für Transkriptionssysteme genannt, die sich an der vertikalen Darstellungsform orientieren. HIAT (Ehlich/Rehbein (1976)) basiert auf der Partiturdarstellungsform. Bloom (1993) beschreibt ein System, das mit der Spaltendarstellung arbeitet.

[18] Die Punkte auf der Zeitachse haben dabei eine vorgegebene Ordnung, was für die Sprecher/Kategroie-Paare zunächst nicht gilt. Es lässt sich aber oft so etwas wie eine kanonische Ordnung der Spalten bzw. Partiturzeilen festlegen, die gewisse Leseerwartungen und –gewohnheiten reflektiert. So werden in einer Partiturdarstellung in der Regel Annotationszeilen direkt unter diejenige Zeile positioniert, die die zugehörigen annotierten Elemente enthält, und bei einer Spaltendarstellung wird der aktivste Sprecher meist in der am weitesten links stehenden Spalte notiert. Edwards (1993: 11) weist sogar darauf hin, dass die Spaltendarstellung aus diesem Grunde besonders für assymetrische Diskurse, d.h. Diskurse, bei denen ein Sprecher wesentlich höhere Gesprächsanteile als alle anderen Sprecher hat, geeignet ist:

> „[...] Column format is useful in highlighting asymmetries among interactants.“

[19] Die Beispiele beziehen sich auf das Beispiel einer Basis-Transkription im vorigen Abschnitt.

[20] Ehlich/Rehbein (1976: 25)

kurse, bei deren Transkription nur *eine* Ereigniskategorie verwendet wird und bei denen die einzelnen Ereignisse zeitlich aufeinander folgen, ohne sich zu überschneiden, funktioniert diese Vorgehensweise problemlos, und die vertikale Darstellung lässt sich aus den in einer Basis-Transkription vorhandenen Informationen konstruieren, z.B.:

Basis-Transkription:

Diskurs	Basis-Transkription B(e)				
E	**D**	**T(Start)**	**T(Ende)**	**S**	**C**
e_1	Du fällst mir immer ins Wort.	t_0	t_1	MAX	verbal
e_2	Stimmt ja gar nicht.	t_1	t_2	TOM	verbal
e_3	Vielleicht hast Du recht.	t_2	t_3	MAX	verbal

zugehörige vertikale Darstellung:

MAX	t_0	t_1	d_1
TOM	t_1	t_2	d_2
MAX	t_2	t_3	d_3
↓	(weitere Sprecher/ Zeitpunkte-Tripel)		

MAX: *Du fällst mir immer ins Wort.*
TOM: *Stimmt ja gar nicht.*
MAX: *Vielleicht hast Du recht.*

Wenn mehrere Ereigniskategorien bei der Transkription verwendet werden, wird in der Regel davon ausgegangen, dass dabei eine Kategorie (meist die „verbale“) den anderen übergeordnet ist[21] und Ereignisse der untergeordneten Kategorien nur zeitgleich zu Ereignissen der übergeordneten Kategorie des selben Sprechers transkribiert werden. So kann immer noch aus den Informationen einer Basis-Transkription eine vertikale Darstellung konstruiert werden, indem die Tripel auf der vertikalen Achse um die Ereigniskategorien ergänzt werden, z.B.:

Basis-Transkription:

Diskurs	Basis-Transkription B(e)				
E	**D**	**T(Start)**	**T(Ende)**	**S**	**C**
e_1	Du fällst mir immer ins Wort.	t_0	t_1	MAX	verbal
e_4	fuchtelt mit den Armen	t_0	t_1	MAX	non-verbal
e_2	Stimmt ja gar nicht.	t_1	t_2	TOM	verbal
e_5	abfällige Geste	t_1	t_2	TOM	non-verbal
e_3	Vielleicht hast Du recht.	t_2	t_3	MAX	verbal

zugehörige vertikale Darstellung[22]:

MAX	t_0	t_1	verbal	d_1
MAX	t_0	t_1	non-verbal	d_4
TOM	t_1	t_2	verbal	d_2
TOM	t_1	t_2	non-verbal	d_5
MAX	t_2	t_3	verbal	d_3
↓	(weitere Sprecher/ Zeitpunkte/Kategorien-Tupel)			

MAX: *Du fällst mir immer ins Wort.*
fuchtelt mit den Armen
TOM: *Stimmt ja gar nicht.*
abfällige Geste
MAX: *Vielleicht hast Du recht.*

Es lässt sich jedoch wohl kaum bestreiten, dass die bis hier gemachten Einschränkungen –

[21] Vgl. hierzu vor allem die Unterscheidung zwischen ‚main tier’ und ‚dependent tiers’ in den CHAT-Konventionen (MacWhinney(1995))
[22] Wie Ereignisbeschreibungen über- und untergeordneter Kategorien dabei letztendlich angeordnet werden – ob, wie im Beispiel hier, untereinander auf separaten Zeilen oder, wie im obigen Beispiel einer vertikalen Darstellung, nebeneinander und durch typographische Mittel (eckige Klammern) voneinander getrennt - ist an dieser Stelle nicht wichtig, kann aber an anderer Stelle durchaus relevant sein. Vgl. hierzu Edwards(1992: 438ff).

„ein Sprecher zur selben Zeit“ und „Ereignisse untergeordneter Kategorien sind zeitgleich zu Ereignissen übergeordneter Kategorien“ – zu stark sind, um in ein angemessenes Modell zur Beschreibung von Diskurstranskriptionen übernommen zu werden. Sie entsprechen nicht der Wirklichkeit spontaner Diskurse, bei denen Sprecher z.B sich gegenseitig ins Wort fallen oder Gesten unabhängig von sprachlichen Handlungen vollziehen. Tatsächlich beschränken sich auch die meisten auf einer vertikalen Darstellungsweise basierenden Transkriptionssysteme nicht in dieser Weise, sondern beinhalten Methoden, um die totale oder partielle Überschneidung von Ereignissen desselben oder unterschiedlicher Sprecher darzustellen. Rehbein et al.(1993: 9ff) diskutieren drei solcher Darstellungsmethoden. Von diesen kann jedoch nur eine aus den in einer Basis-Transkription enthaltenen Informationen berechnet werden: bei dieser Methode werden Beschreibungen simultaner Ereignisse zwar in aufeinanderfolgenden Zeilen aufgeführt, es wird jedoch typographisch (meist durch eine Klammerung) deutlich gemacht, dass an dieser Stelle der zeitliche Fluss angehalten wird, d.h. die in den beiden Zeilen dargestellten Ereignisse zeitgleich stattfinden, z.B.:

Diskurs	Basis-Transkription B(e)				
E	**D**	**T(Start)**	**T(Ende)**	**S**	**C**
e_1	Du fällst mir immer ins	t_0	t_1	MAX	verbal
e_2	Wort.	t_1	t_2	MAX	verbal
e_3	Siehst Du, Du hast es schon wieder getan.	t_3	t_4	MAX	verbal
e_4	Stimmt	t_1	t_2	TOM	verbal
e_5	ja gar nicht.	t_2	t_3	TOM	verbal

MAX	t_0	t_1	verbal	d_1
MAX	t_1	t_2	verbal	d_2
TOM	t_1	t_2	verbal	d_4
TOM	t_2	t_3	verbal	d_5
MAX	t_3	t_4	verbal	d_3
↓	(weitere Sprecher/ Zeitpunkte/Kategorien-Tupel)			

MAX: *Du fällst mir immer ins*
MAX: *<Wort>*
TOM: *<Stimmt>*
TOM: *ja gar nicht.*
MAX: *Siehst Du, Du hast es schon wieder getan.*

Wie Rehbein et al. (1993) bemerken, hat dieser Darstellungstyp jedoch den Nachteil, dass

> „ein [Sprecher-]Beitrag [...] über mehrere Zeilen hinwegspringend gelesen werden [muss].“[23],

was der gängigen Lesegewohnheit, wie sie sich z.B. aus Theaterskripten oder in Zeitungen abgedruckten Interviews etabliert hat, zuwiderläuft und das Transkript deshalb für den menschlichen Benutzer schwerer lesbar macht.
Meist werden deshalb mehrere einem Sprecher zugeordnete Ereignisse zu Sprecherbeiträgen zusammengefasst und können so auf einer Zeile belassen werden. Sich überlappende Ereignisse werden dabei wiederum typographisch kenntlich gemacht[24], z.B.:

[23] Rehbein et al. (1993: 11)
[24] Dies entspricht einer weiteren der in Rehbein et al. (1993) diskutierten drei Verfahrensweise. Bei der verbleibenden werden die Überlappungen zusätzlich an einer gemeinsamen vertikalen Position aufgereiht (d.h. in der Darstellung wird der entsprechende Sprecherbeitrag passend eingerückt). Die Autoren bemerken hierzu (S. 10):

> „Hier wird eine unsystematische ad-hoc-Lösung zur Darstellung von Überlappungen gewählt, die nicht standardisiert werden kann. Man verwendet generell zwar die Zeilenschreibweise [i.e. die vertikale Darstellung, T.S.], geht aber nach Bedarf [...] von ihr ab, indem man den *Beginn der Zeile* nicht mehr linksbündig, sondern relativ zu einer vorhergehenden lokalisiert [...] so wird bei Überlappungen, die mehr als eine Zeile in Anspruch nehmen, diese Darstellung nicht mehr möglich [...]“

Dieser Einwand ist gerechtfertigt. Diese dritte Verfahrensweise ist daher für ein System, bei dem die Darstellung aus einer Inhaltsrepräsentation berechnet werden soll, deshalb denkbar ungeeignet – eine solche Berechnung ist in manchen Fällen schlicht und einfach nicht möglich.

MAX	t_0	t_2	verbal	d_1 d_2
TOM	t_1	t_3	verbal	d_4 d_5
MAX	t_3	t_4	verbal	d_3
↓	(weitere Sprecher/Zeitpunkte/Kategorien-Tupel)			

MAX: *Du fällst mir immer ins<Wort>.*
TOM: *<Stimmt> ja gar nicht.*
MAX: *Siehst Du, Du hast es schon wieder getan.*

An dieser Stelle ist die in einer Basis-Transkription enthaltene Information nicht mehr ausreichend, um die zugehörige Darstellung zu berechnen – es fehlt die Information, welche Ereignisse zu größeren Einheiten (in der Regel handelt es sich dabei um Turns oder Äußerungen) zusammengefasst werden können und müssen[25]. Im nächsten Abschnitt wird dargestellt werden, warum solche Information nicht nur darstellungstechnisch relevant ist, sondern auch für andere Aspekte der Verarbeitung von Diskurstranskriptionen benötigt wird.

4. Zeitliche und sprachliche Struktur

Die in Abschnitt 2 beschriebene Basis-Transkription stützt sich in erster Linie auf die *zeitliche* Struktur eines Diskurses. Die Einteilung des Diskurses in Ereignisse und damit die der Transkription zugrundegelegte Zeitachse kann in der Regel nach rein zeitlichen Kriterien erfolgen - beispielsweise können die Handlungen eines Sprechers so in Ereignisse unterteilt werden, dass immer dann, wenn sich ein Wechsel in der momentanen Sprecherkonstellation ergibt (d.h. z.B. ein Sprecher hinzukommt oder wegfällt), der Zeitachse ein neuer Zeitpunkt hinzugefügt wird. Wie im vorigen Abschnitt gezeigt, reicht eine solche Strukturierung für eine angemessene Darstellung des Diskurses in seinem zeitlichen Verlauf bereits aus.
Eine Diskurstranskription dient jedoch zwar zuallererst, aber nicht alleine diesem Zweck. Sie soll ihrem Benutzer nämlich außer der zeitlichen Struktur des Diskurses auch dessen *sprachliche* Struktur verdeutlichen. Implizit kann der Betrachter diese Struktur oft bereits teilweise aus den Zeichenketten, die für die Ereignisbeschreibung verwendet werden, erschließen. So lässt sich in den oben verwendeten Beispielen erkennen, dass manche (verbalen) Ereignisse aus mehreren Wörtern zusammengesetzt sind oder sich zu größeren sprachlichen Einheiten zusammenfassen lassen.
Es wurde jedoch ebenfalls bereits gezeigt, dass eine solche implizite sprachliche Strukturierung nicht für alle Zwecke ausreichend ist – um z.B. eine bestimmte Form der vertikalen Darstellung berechnen zu können, muss in der computerseitigen Datenrepräsentation explizit vermerkt sein, welche Ereignisse sich zu Sprecherbeiträgen kombinieren lassen. Noch viel wichtiger werden sprachliche Strukturelemente, wenn ausgehend von einer Basis-Transkription eine Weiterverarbeitung in Form einer oder mehrerer Annotationen[26] erfolgen

[25] Edwards (1993: 16ff) unterscheidet in diesem Zusammenhang ‚event-based systems' und ‚utterance-based systems'
[26] Anders als Bird/Liberman (2001: 3) und zahlreiche andere Autoren möchte ich die Begriffe Transkription (bzw. *transcription*) und Annotation (bzw. *coding*) unterscheiden, etwa im Sinne von MacWhinney (1995):

> „It is important to recognize the difference between *transcription* and *coding* [i.e.: Annotation, T.S.]. Transcription focuses on the production of a written record that can lead us to understand, albeit only vaguely, the flow of the original interaction. Transcription must be done directly off an audiotape or, preferably, a videotape. Coding, on the other hand, is the process of recognizing, analyzing, and taking note of phenomena in transcribed speech. Coding can often be done by referring only to a written transcript."

Zwar führen Bird/Liberman berechtigte Argumente an, die hier als *transcription* und *coding* definierten Begriffe unter dem Terminus *annotation* zusammenzufassen, diese Verallgemeinerung ist aber nur dann zu rechtfertigen, wenn man eine Transkription als ein gegebens Datum betrachtet und die Art und Weise seiner Entstehung außer Acht lässt. Betrachtet man das Transkribieren und Annotieren hingegen als praktischen Prozess, ist es nach meiner Auffassung durchaus angemessen, die beiden zweifellos verschiedenen Aktivitäten auch mit verschiedenen Begriffen zu bezeichnen.

soll, denn gerade solche Annotationen sind oft für eine automatisierte Analyse von Transkriptionsdaten entscheidend. In aller Regel werden solchen Annotationen bestimmte sprachliche Einheiten zugrundegelegt, z.B.:

- bezieht sich eine *Übersetzung* in der Regel auf *Äußerungen* oder *Wörter*
- bezieht sich ein *Part-Of-Speech-Tagging* in der Regel auf Wörter
- bezieht sich eine *Phrasenstruktur-Annotation* auf ein oder mehrere *Wörter* bis hin zu einer ganzen *Äußerung*
- usw.

Anders als bei der relativ objektiv feststellbaren zeitlichen Strukturierung eines Diskurses herrscht über die Einheiten, die bei der sprachlichen Strukturierung eines Diskurses zu verwenden sind, aber keineswegs unbedingt Einverständnis zwischen verschiedenen Forschergruppen, wie z.B. folgendes Zitat zeigt:

> "Now, TEI specifies an utterance. It's a defined concept and there are certain things you can't do, like you can't interrupt it. Now, I believe in life that you can interrupt utterances. I don't have utterances as units in my descriptive system, and indeed many transcription systems don't have rigorously defined utterances. [...] I will personally not accept TEI if it requires me to have an utterance under the definition that Lou Bernard was using, because that is far too rigorous for me and it doesn't represent the world, as far as I'm concerned."[27]

Abgesehen davon, dass der hier vorgebrachte Einwand sachlich nicht ganz richtig ist (das TEI-System sieht durchaus Methoden vor, das Unterbrechen einer Äußerung zu kodieren), zeigt er, dass ein Transkriptionssystem, das bestimmte festgelegte sprachliche Einheiten verwendet, seine Universalität einbüßt, weil diese Festlegung u.U. mit den sprachtheoretischen Annahmen seiner potentiellen Benutzer nicht vereinbar ist – gewisse Kategorien oder Einheiten sind in verschiedenen Systemen verschieden definiert oder gar nicht vorhanden.

Ein Transkriptionssystem, das der Anforderung genügen will, verschiedene vorhandene Systeme zu vereinen, kann deshalb zunächst nur von der Annahme ausgehen, dass Diskurstranskriptionen und –annotationen außer mit zeitlichen Einheiten auch mit sprachlichen Einheiten operieren. Um welche Einheiten es sich dabei handelt, kann es jedoch aus den oben genannten Gründen nicht festlegen.

Wenn mehrere sprachliche Einheiten verwendet werden, stehen diese weiterhin in der Regel nicht unverbunden nebeneinander, sondern sind oft in hierarchischen Ordnungen organisiert[28]. Beispielsweise wird in vielen Systemen angenommen, dass sich die Einheit ‚Turn' aus einer oder mehreren Äußerungen und die Einheit ‚Äußerung' aus einem oder mehreren Wörtern zusammensetzt. Eine solche hierarchische Ordnung gilt jedoch nicht zwingend für alle benutzten sprachlichen Einheiten. Insbesondere ist es möglich, dass diese sich in mehreren, sich teilweise überschneidenden Hierarchien organisieren. Ein Beispiel für ein System, bei dem zwei sich überlappende Hierarchien verwendet werden, findet sich in Isard (2001). Hier wird auf der sprachlichen Ebene einerseits eine Hierarchie

[27] Sinclair (1995: 108)
[28] vgl. hierzu auch Bird/Liberman(2001: 15)

> „Existing annotated speech corpora always involve a hierarchy of some kind, even if they do not focus on very elaborate types of linguistic structure."

```
Dialogue → Move → Word
```

und andererseits eine Hierarchie

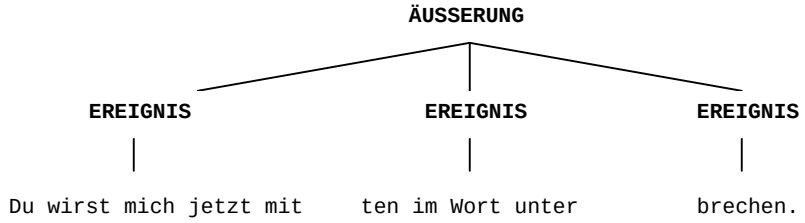
```
Disfluency → Reparandum / Repair → Word
```

verwendet, die sich die unterste Ebene – und nur diese – teilen.

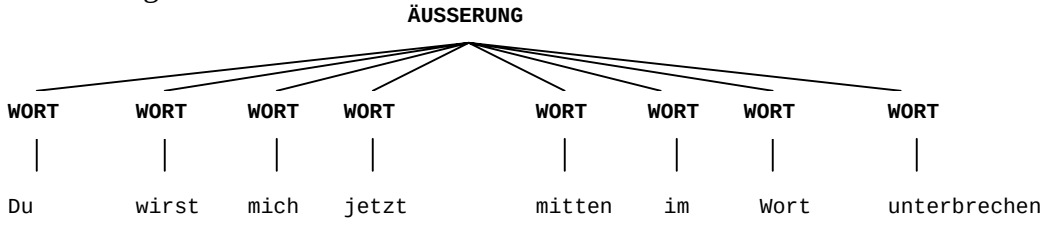
Schließlich stehen auch zeitliche und sprachliche Struktur einer Diskurstranskription nicht unverbunden nebeneinander, sondern beziehen sich aufeinander. So wurde oben angedeutet, wie in der vertikalen Darstellung mehrere zeitliche Einheiten (‚Ereignis') für eine bessere Lesbarkeit des Transkriptes zu sprachlichen Einheiten (z.B. ‚Sprecherbeitrag', ‚Turn' oder ‚Äußerung') *zusammengefasst* werden, oder wie für eine Annotation zeitliche Einheiten in mehrere sprachliche Einheiten (z.B. ‚Wort') *zerlegt* werden können. Auch hier kann jedoch nicht – wie die Wörter ‚Zusammenfassen' und ‚Zerlegen' vielleicht irrtümlicherweise nahe legen – von einer hierarchischen Beziehung ausgegangen werden. Einerseits können zwar zeitliche Strukturen unterhalb jeder sprachlichen Strukturebene relevant sein, andererseits kann aber in der Transkriptionspraxis die Möglichkeit, auch die kleinsten sprachlichen Einheiten untereinander zeitlich zu strukturieren, nicht unbedingt immer wahrgenommen werden. Ein Beispiel soll dies verdeutlichen:

```
MAX: Du wirst mich jetzt mit[ten im Wort unter]brechen.
TOM:                         [Da hast Du recht.]
```

Wie durch die Klammerung und die horizontale Anordnung angedeutet, liegt hier eine teilweise Überlappung von Maxens Äußerung durch Toms Äußerung vor. Die zeitliche Struktur dieses Diskursausschnittes kann beschrieben werden, indem Maxens Äußerung in drei zeitliche Einheiten („Ereignisse" im oben definierten Sinne) unterteilt wird:

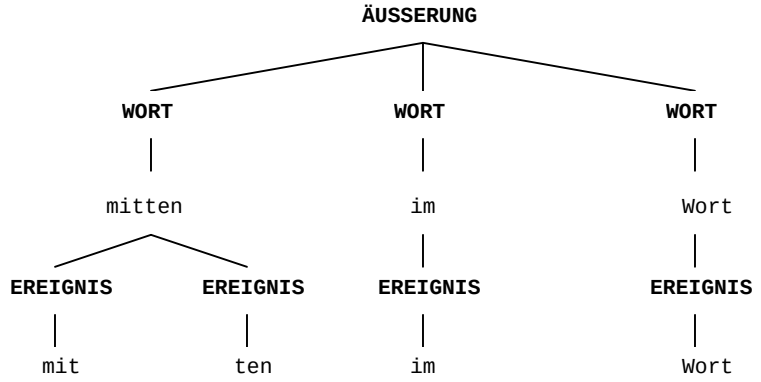


Auf der sprachlichen Strukturebene hingegen ist zum Beispiel eine Unterteilung der Äußerung in Wörter naheliegend:



Es lässt sich auf Anhieb erkennen, dass die so verwendeten Einheiten ‚Ereignis' und ‚Wort' nicht in eine hierarchische Ordnung zu bringen sind – es kommt sowohl vor, dass ein Ereignis sich aus mehreren Wörtern zusammensetzt („*Du wirst mich...*") als auch, dass sich ein Wort

über mehrere Ereignisse verteilt („*unterbrechen*“). Eine theoretisch denkbare Lösung für dieses Problem wäre die Beschränkung, dass zeitliche Einheiten nicht mehr als ein Wort umfassen dürfen[29], etwa:



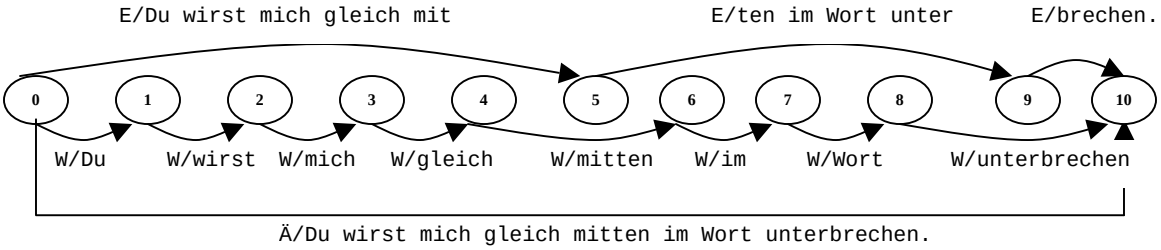
Eine solche Beschränkung ist jedoch in zweierlei Hinsicht problematisch:
Erstens wird so auch die Basis-Transkription von sprachlichen Einheiten abhängig, weil ja bei der Bestimmung zeitlicher Diskurs-Einheiten sprachliche Kategorien zugrundegelegt werden. Wie oben dargestellt würde dadurch die Basis-Transkription in höherem Maße theorieabhängig und das Transkriptionssystem damit weniger universell.
Zweitens ergeben sich schwerwiegende Probleme beim praktischen Transkribieren: im obigen Beispiel überlappen sich zwei Ereignisse, die jeweils mehrere Wörter umfassen, d.h. die bei der angesprochenen Vorgehensweise auch jeweils aus mehreren zeitlichen Einheiten bestehen würden. Wie in Abschnitt 2 und 3 dargelegt, müssten diese zeitlichen Einheiten für eine adäquate Basis-Transkription in eine *gemeinsame* Zeitachse eingeordnet werden, d.h. der Transkribent müsste (mindestens) die in der Überlappung vorkommenden Wörter *beider* Sprecher zeitlich ordnen. Es kann davon ausgegangen werden, dass ein solches Maß an Genauigkeit den Transkriptionsprozess nicht nur enorm (und unnötig) verkompliziert, sondern dass ein solches Vorgehen auch an kognitive Grenzen beim Transkribenten stößt, d.h. dass er unter Umständen gar nicht in der Lage ist, eine derart feine zeitliche Einordnung vorzunehmen[30].

Es stellt sich daher die Frage, wie ein Transkriptionssystem für den Computer Möglichkeiten zur Verfügung stellen kann, zeitliche und sprachliche Struktur einer Diskurstranskription zu kodieren, ohne einerseits zu starke Annahmen bezüglich der zu verwendenden sprachlichen Einheiten und deren Beziehungen untereinander und zu den zeitlichen Einheiten zu machen, und ohne andererseits eventuell trotzdem vorhandene spezifische Strukturinformationen wie hierarchische Ordnungen zu verlieren.

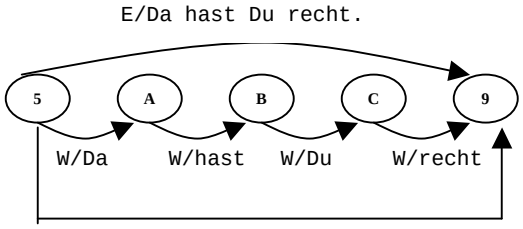
Der in Bird/Liberman (2001) beschriebene Annotationsgraphen-Formalismus bietet eine Lösung für einen Teil dieses Problems an: sprachliche Strukturen werden hier zunächst auf zeitliche Strukturen zurückgeführt. Die zeitliche Strukturierung ist dabei nicht auf Hierarchien angewiesen (sie zeichnet sich vielmehr durch das aus, was die Autoren „sequential“ bzw. „parallel structure“[31] nennen), aber hierarchische Strukturen lassen sich aus zeitlichen rekonstruieren. So könnte Maxens Äußerung im obigen Beispiel etwa wie folgt als Annotations-

[29] Eine derart eingeschränkte Definition eines Ereignisses liegt z.B. den in Isard (2001) beschriebenen ‚timed units’ zugrunde.
[30] Üblicherweise wird auch in den Transkriptionskonventionen nur verlangt, *Anfang und Ende* einer Überlappung zu kennzeichnen. Vgl. hierzu z.B. Ehlich/Rehbein (1976: 27), Selting et al.(1998: 97f) und MacWhinney(1995: 74ff).
[31] vgl. Bird/Liberman (2001: 12)

Graph beschrieben werden:



Zeitliche Strukturen (E(reignisse)) und sprachliche Strukturen (Ä(ußerungen) und W(örter)) können hier sinnvoll gemeinsam kodiert werden, weil vom Konzept einer Hierarchie nicht direkt Gebrauch gemacht wird. Sofern Hierarchien vorhanden sind, lassen sie sich aber einfach rekonstruieren[32] – z.B. sind die Zeitpunkte 1 und 2, die durch das Wort ‚*Du*' verbunden sind, sowohl in der Zeitspanne zwischen den Zeitpunkten 0 und 5, die durch das Ereignis ‚*Du wirst mich gleich mit*' verbunden sind, als auch in der Zeitspanne zwischen den Zeitpunkten 0 und 10, die durch die gesamte Äußerung verbunden sind, voll enthalten, und diese zeitliche Beziehung lässt sich als hierarchische Beziehung interpretieren: das Wort ‚*Du*' ist sowohl Teil des Ereignisses als auch der Äußerung.
Mit der zweiten Äußerung aus dem Beispiel würde im Annotationsgraphen-Formalismus analog verfahren werden:



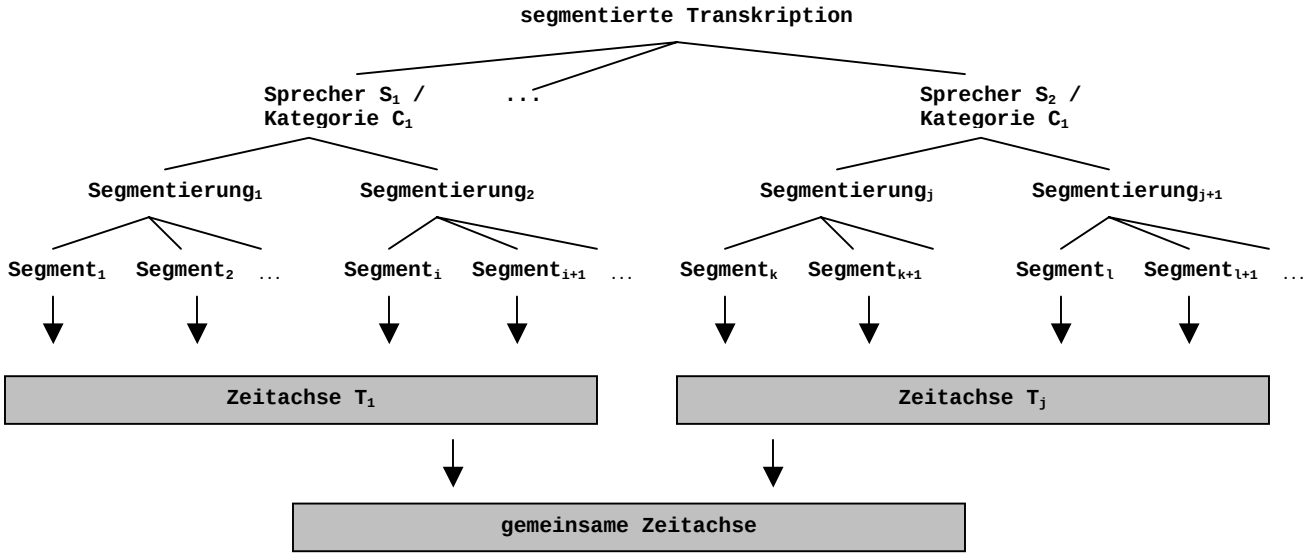
Die Zeitpunkte 5 und 9 können dabei aus dem obigen Teilgraphen übernommen werden, womit die Überlappung der Ereignisse repräsentiert ist. Für die Beschreibung der Wortstruktur müssen hingegen weitere Zeitpunkte hinzugefügt werden. Dass die Zeitachse, die sich so ergibt, lediglich partiell geordnet ist (d.h. dass z.B. für die Zeitpunkte 7 und B keine Ordnung existiert), wird im Annotationsgraphen-Formalismus ausdrücklich in Kauf genommen. Wie oben angemerkt, ergeben sich dadurch zwar Probleme bei den herkömmlichen Darstellungsmethoden (die alle auf eine für die gesamte Transkription gültige Zeitachse angewiesen sind). Da die fehlende Information aber aus ebenfalls oben dargelegten Gründen einfach nicht in jedem Falle vom Transkribenten geliefert werden kann, beinhaltet dieses Modell keinerlei Reduktionen, die nicht schon à priori vorhanden waren, und wird deshalb als Grundlage des hier beschriebenen Transkriptionssystems übernommen.

Sprachliche Strukturen werden also auf zeitliche Strukturen abgebildet. Dadurch können zur

[32] Bird/Liberman (2001: 15) drücken dies wie folgt aus:

> „[…] if the substring spanned by arc a_i properly contains the substring spanned by a_j then the constituent corresponding to a_i must dominate the constituent corresponding to a_j […] Hierarchical relationships are encoded only to the extent that they are implied by this graph-wise inclusion."

formalen Repräsentation dieselben Mechanismen wie bei der Definition der Basis-Transkription verwendet werden, d.h. eine solche Transkription kann wiederum als Zuordnung von Teilen des Diskurses (bzw. der Aufnahme) zu einer Zeitachse, einem Sprecher etc. betrachtet werden. Da die Einteilung des Diskurses dabei nach sprachlichen und zeitlichen Kriterien erfolgen kann, erscheint es sinnvoll, nicht mehr von *Ereignissen*, sondern allgemeiner von *Segmenten* zu sprechen; für die Transkription bietet sich dann der Terminus *segmentierte Transkription* an. Eine solche segmentierte Transkription kann also als eine erweiterte Basis-Transkription betrachtet werden, der eine Strukturebene hinzugefügt wird – Segmente eines Sprechers und einer Kategorie werden in verschiedenen *Segmentierungen* organisiert, und für jede Segmentierung existiert eine individuelle Zeitachse, wobei sich aber die verschiedenen individuellen Zeitachsen Zeitpunkte teilen können. Diejenigen Zeitpunkte, die in allen individuellen Zeitachsen vorkommen, bilden die gemeinsame Zeitachse:



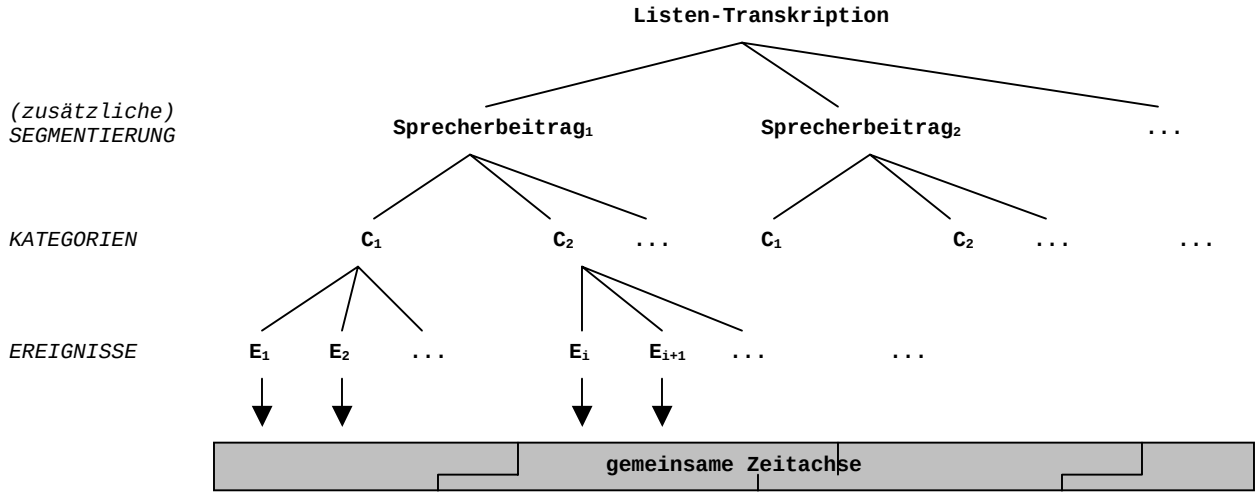
Die Beispiel-Basis-Transkription aus Abschnitt 2 ließe sich demnach z.B. wie folgt durch eine Segmentierung in Äußerungen und Wörtern erweitern:

```
Segmente = {s1, s2, s3, …}
Segmentierungen = {Ereignis, Wort, Äußerung}
T = {t0, t1, t2, t3, t4}     t0 < t1 <t2 < t3 < t4                                  (gemeinsame Zeitachse)
T1 = {t0, t5, t6, t7, t8, t9, t10, t11, t12 t1, t2, t3, t4}    t0 < t5 <t6 < ... < t3 < t4   (individuelle Zeitachse)
T2 = ...                                                                            (individuelle Zeitachse)
D = {“Du fällst mir immer ins“, „Wort.“, ..., „abfällige Handbewegung“,             (Beschreibungen)
     “Du fällst mir immer ins Wort.“, ...,
     “Du“, “fällst“, “mir“, “immer“, “ins“, “Wort“, ...}
S = {MAX,TOM}                                                                       (Sprecher)
C = {verbal, non-verbal}                                                            (Kategorien)
```

Diskurs	segmentierte Transkription					
Segmente	**D**	**T(Start)**	**T(Ende)**	**S**	**C**	**Segmentierung**
s_1	Du fällst mir immer ins	t_0	t_1	MAX	verbal	Ereignis
s_2	Wort.	t_1	t_2	MAX	verbal	Ereignis
s_3	Siehst Du, Du hast es schon wieder getan.	t_3	t_4	MAX	verbal	Ereignis
s_8	Du fällst mir immer ins Wort.	t_0	t_2	MAX	verbal	Äußerung
s_9	Siehst Du, Du hast es schon wieder getan.	t_3	t_4	MAX	verbal	Äußerung
s_{10}	Du	t_0	t_5	MAX	verbal	Wort
s_{11}	fällst	t_6	t_7	MAX	verbal	Wort
s_{12}	mir	t_8	t_9	MAX	verbal	Wort
s_{13}	immer	t_{10}	t_{11}	MAX	verbal	Wort
s_{14}	ins	t_{12}	t_1	MAX	verbal	Wort
s_{15}	Wort	t_1	t_2	MAX	verbal	Wort
…	…	…	…	…	…	…
s_4	fuchtelt mit den Armen	t_0	t_2	MAX	non-verbal	Ereignis
s_5	Stimmt	t_1	t_2	TOM	verbal	Ereignis
s_6	ja gar nicht.	t_2	t_3	TOM	verbal	Ereignis
s_7	abfällige Handbewegung	t_1	t_3	TOM	non-verbal	Ereignis

Statt die segmentierte Transkription als eine Erweiterung der Basis-Transkription zu betrachten, kann auch umgekehrt die Basis-Transkription als eine Einschränkung der segmentierten Transkription angesehen werden: wenn für jeden Sprecher und jede Kategorie in einer segmentierten Transkription nur eine Segmentierung vorhanden ist und diese sich ausschließlich auf die gemeinsame Zeitachse bezieht, können dieselben Informationen auch in einer Basis-Transkripion kodiert werden. Die Menge der Basis-Transkriptionen ist somit eine Teilmenge der Menge der segmentierten Transkriptionen. Dies ist zum einen auf der Inhaltsseite des Systems relevant, denn es bedeutet, dass eine Basis-Transkiption immer auch bereits eine segmentierte Transkription ist und somit im Prinzip nur ein Format benötigt wird. Es ist aber zum anderen auch für die Darstellungsseite des Systems von Interesse, denn es ist genau die Teilmenge der Basis-Transkriptionen, die sich mit den herkömmlichen Darstellungstypen der Partitur- und Spaltendarstellung darstellen lässt.
Die Menge der im vertikalen Darstellungstyp darstellbaren Transkriptionen schließlich ist ebenfalls eine Teilmenge der segmentierten Transkriptionen sowie eine Obermenge der Basis-Transkriptionen: bei dem in Abschnitt 3 angesprochenen „Zusammenfassen“ mehrerer Ereignisse zu Sprecherbeiträgen handelt es sich schlicht um genau eine Segmentierung, die den Ereignissen einer Basis-Transkription übergeordnet ist. „Übergeordnet“ bedeutet in diesem Zusammenhang einerseits, dass sich ein Ereignis immer *genau einem* Sprecherbeitrag zuordnen lässt[33] und andererseits, dass diese Segmentierung kategorienübergreifend gilt bzw. für alle Ereigniskategorien *eines Sprechers* identisch erfolgt. Es ergeben sich daher keine neuen sprecher- oder kategorienspezifischen Zeitpunkte, d.h. eine solche Transkription (die im folgenden Listen-Transkription genannt werden wird), kommt wie eine Basis-Transkription mit einer gemeinsamen Zeitachse für alle Sprecher und alle Ereigniskategorien aus. Die Segmentierung in Sprecherbeiträge kann als ein weiteres Strukturelement angesehen werden, wodurch die Strukturierung in „Tiers“ etc. auf die Unterscheidung verschiedener Ereigniskategorien reduziert wird[34]:

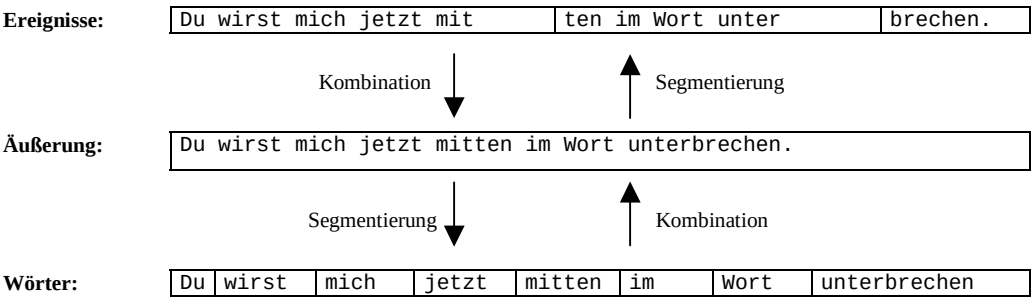
[33] Die Segmentierungen bilden also eine Hierarchie **Sprecherbeitrag → Ereignis**
[34] Der Begriff „Tier“ wird in den CHAT-Konventionen (MacWhinney (1995)), die ja am vertikalen Darstellungstyp orientiert sind, auch in genau dieser Weise gebraucht



Wie die Grafik andeutet, lassen sich in diesem Falle, anders als bei der segmentierten und der Basis-Transkription, bereits die obersten Strukturelemente in eine (zeitliche) Ordnung bringen. Wenn die Segmentierung in Sprecherbeiträge der Ereignissegmentierung nicht über- sondern „gleichberechtigt“ beigeordnet wird, ist eine solche Listen-Transkription bereits in eine segmentierte Transkription überführt. Durch Entfernen der Segmentierung in Sprecherbeiträge und einem neuerlichen Zusammenfassen von Sprechern und Ereigniskategorien ergibt sich hingegen eine Basis-Transkription.

5. Ereigniskategorien und -typen

Im vorigen Abschnitt wurde gezeigt, wie die Strukturierung einer Diskurstranskription nach zeitlichen Einheiten mit ihrer Strukturierung nach sprachlichen Einheiten zusammenhängt. Dieser Zusammenhang lässt sich mit dem Begriff ‚Segmentierung’ beschreiben – durch Kombinieren und Segmentieren von Zeichenketten, die zeitliche Ereignisse beschreiben, ergeben sich Beschreibungen sprachlicher Segmente, z.B.:



Bei einer solchen Segmentierung oder Kombination wird ausgenutzt, dass die Verkettung von Beschreibungen zweier aufeinanderfolgender Ereignisse eine Zeichenkette ergibt, die in sinnvoller Weise die Gesamtheit dieser beiden Ereignisse beschreibt und dass umgekehrt jeder Teil einer Ereignisbeschreibung in sinnvoller Weise einen Teil des Ereignisses beschreibt. Diese Eigenschaft besitzen jedoch nicht alle denkbaren Ereignisbeschreibungen. So wird z.B. in der Beispiel-Transkription aus Abschnitt 2 ein non-verbales Ereignis mit der Zeichenkette

```
fuchtelt mit den Armen
```

beschrieben. Wenn auch das so beschriebene Ereignis als solches sicherlich segmentierbar ist, so gilt hier jedoch nicht, dass ein Teil des Ereignisses durch einen Teil der Zeichenkette, die zur seiner Beschreibung verwendet wurde, sinnvoll beschrieben werden kann - das Ereignis setzt sich z.B. nicht aus zwei Teilereignissen zusammen, die sich mit den Zeichenketten ‚fuch' bzw. ‚telt mit den Armen' beschreiben ließen. Vielmehr wird jedes Teilereignis durch die gesamte Zeichenkette adäquat beschrieben, d.h. das Ereignis setzt sich z.B. aus zwei Teilereignissen zusammen, die sich beide mit der Zeichenkette ‚fuchtelt mit den Armen' beschreiben lassen.
Bei einem weiteren Typ von Beschreibungen ist eine Segmentierung zwar u.U. möglich, aber weder theoretisch noch praktisch sinnvoll. Man betrachte dazu das folgende Beispiel:

	0	1	2
MAX:	Du fällst mir immer	ins Wort	
	You're always interrupting me.		
TOM:		Stimmt ja	gar nicht.
		That's not true.	

Die kursiv gedruckten Beschreibungen sind Annotationen. Sie beziehen sich nicht unmittelbar auf tatsächlich im Diskurs vorhandene Ereignisse, sondern zunächst auf andere Beschreibungen und damit nur mittelbar auf die Zeitachse. Sie können also ihrer Natur nach nur Zeitpunkten zugeordnet werden, denen bereits andere (nicht annotierende) Segmente zugeordnet sind, wodurch sich eine Segmentierung wie die oben beschriebene erübrigt[35].
Um eine Segmentierung wie die oben angedeutete automatisch vornehmen zu können, muss also wiederum explizite Information gegeben werden, welche Ereignisse für eine Segmentierung geeignet sind und welche nicht.
Aus den angeführten Beispielen wird deutlich, dass diese Eigenschaft in irgendeiner Weise mit den Ereigniskategorien zusammenhängt – verbale Ereignisse werden in der Regel so beschrieben, dass die verwendeten Zeichenketten segmentier- und kombinierbar sind, bei nonverbalen Ereignissen ist dies in der Regel nicht der Fall, und Übersetzungen sind, da sie Annotationen sind, ebenfalls nicht für eine Segmentierung geeignet. Um diese Information in das Transkriptionssystem zu integrieren, bieten sich zwei Möglichkeiten an:

> **1.** Das System stellt eine *feste Anzahl von Ereigniskategorien*, deren Verhalten bzgl. Segmentierung und Kombination bekannt ist, zur Verfügung.
>
> **2.** Das System lässt, wie in den obigen Definitionen vorgesehen, die Ereigniskategorien offen, lässt diese aber vom Benutzer bzgl. ihres Segmentierungsverhaltens *typisieren*, d.h. die Datenstruktur sieht vor, dass jeder benutzten Ereigniskategorie ein Segmentierungstyp zugeordnet wird.

Viele vorhandene Systeme operieren mit festgelegten Ereigniskategorien. Vergleicht man zwei solcher Systeme stellt man jedoch schnell fest, dass sich nur schwer ein gemeinsamer Nenner finden lässt, denn die benutzten Kategorien spiegeln oft spezifische Zielsetzungen der Transkription wider[36]. Im hier beschriebenen System wird deshalb die zweite Möglichkeit gewählt. Es werden folgende Typen von Ereigniskategorien unterschieden:

---

[35] Sie ist auch nicht bei allen Formen der Annotation denkbar – man betrachte z.B. statt der angeführten Übersetzung eine Annotation der Art ‚Aussagesatz'. Eine *Kombination* solcher Beschreibungen ist hingegen in der Regel möglich (und oft auch sinnvoll), da sich dabei ja auf keinen Fall neue Zeitpunkte ergeben.
[36] Vgl. z.B. CHAT (MacWhinney (1995)) und HIAT (Ehlich/Rehbein (1976))

a) Transkription (Typ ‚T') – Ereignisbeschreibungen vom Typ ‚T' sind im oben illustrierten Sinne segmentier- und kombinierbar, d.h. es gilt:

Wenn
$T = \{t_0, t_2\}$ $(t_0<t_2)$ ein Ausschnitt aus der Zeitachse einer Basistranskription ist, die Zeichenkette $d = d_1 \oplus d_2$ die Konkatenation zweier Zeichenketten d_1 und d_2 ist,
$b(e) = (d, t_0, t_2, s, c)$ eine Zuordnung einer Basistranskription ist, die ein Diskurs-Ereignis e angemessen beschreibt, und die Kategegorie c vom Typ ‚T' ist,
dann
gibt es einen Zeitpunkt t_1 mit $t_0<t_1<t_2$ und eine Unterteilung von e in Teil-Ereignisse e_1 und e_2, so dass

$$b(e_1) = (d_1, t_0, t_1, s, c) \quad \text{und}$$
$$b(e_2) = (d_2, t_1, t_2, s, c)$$

angemessene Beschreibungen der Teilereignisse sind[37]. Die Umkehrung gilt ebenso.

Üblicherweise findet sich in jeder Transkription nur eine Ereigniskategorie dieses Typs – nämlich diejenige, in der das verbale Handeln der Sprecher beschrieben wird (z.B. Spur mit „verbaler Kommunikation" bei HIAT, „main tier" bei CHAT)[38].

b) Deskription (Typ 'D') – Deskriptionen können zeitlich unabhängig von anderen Elementen sein, sind aber im Gegensatz zu Transkriptionen *atomar*, d.h. Ereignisbeschreibungen vom Typ ‚D' sind nicht im oben definierten Sinne segmentierbar, es gilt vielmehr:

Wenn
$T = \{t_0, t_2\}$ $t_0<t_2$ ein Ausschnitt aus der Zeitachse einer Basistranskription ist,
d eine Zeichenkette ist,
$b(e) = (d, t_0, t_2, s, c)$ eine Zuordnung einer Basistranskription ist, die ein Diskurs-Ereignis e angemessen beschreibt und die Kategegorie c vom Typ ‚D' ist,
dann
gilt für jeden Zeitpunkt t_1 mit $t_0<t_1<t_2$ und jede Unterteilung von e in Teil-Ereignisse e_1 und e_2, dass

$$b(e_1) = (d, t_0, t_1, s, c) \quad \text{und}$$
$$b(e_2) = (d, t_1, t_2, s, c)$$

angemessene Beschreibungen der Teilereignisse e_1 und e_2 sind. Die Umkehrung gilt ebenso.

[37] Streng genommen gilt diese Eigenschaft in der Praxis nur annäherungsweise. Beispielsweise finden sich in *orthographischen* Transkriptionen gesprochener Sprache Zeichenketten, die nicht unbedingt an jeder, sondern nur an den meisten Stellen sinnvoll segmentierbar sind, z.B. kann die im obigen Beispiel verwendete Zeichenkette ‚gar nicht' nicht in zwei Zeichenketten ‚gar nic' und ‚ht' zerlegt werden, die in sinnvoller Weise Teilereignisse beschreiben würden. Für die praktische Verarbeitung ist eine solche annähernde Segmentierbarkeit aber meiner Einschätzung nach ausreichend. Ein Beispiel für eine an ausnahmslos jeder Stelle segmentierbare Ereignisbeschreibung wäre eine *phonemische* Transkription gesprochener Sprache, die aber in der Praxis selten verwendet wird.

[38] Es gibt meines Wissens auch kein Zeichensystem, das nicht-verbale Handlungen auf im obigen Sinne segmentierbare Zeichenketten abbildet.

c) Annotation (Typ ‚A') – für Beschreibungen vom Typ 'A' gilt:

Wenn
$T = \{t_0, t_{1,\ldots} t_n\}$ $t_0<t_1<\ldots<t_n$ ein Ausschnitt aus der Zeitachse einer Basistranskription ist,
$b(e_a) = (d_a, t_0, t_n, s, c)$ eine Zuordnung der Basistranskription ist
und die Kategegorie c vom Typ ‚A' ist,
dann
gibt es in der Basistranskription eine oder mehrere weitere Zuordnungen

$$b(e_1) = (d_1, t_0, t_1, s, c')$$
$$\ldots$$
$$b(e_n) = (d_n, t_{n-1}, t_n, s, c')$$

so dass d_a eine angemessene Annotation von

$$d' = d_1 \oplus d_2 \oplus \ldots \oplus d_n$$

ist.

d) Datei-Verweis (Typ 'L')

Bis hierher wurde unausgesprochen davon ausgegangen, dass Ereignis- oder Segmentbeschreibungen immer in Form von Zeichenketten erfolgen. Es ist jedoch denkbar – und kommt in technisch fortgeschritteneren Transkriptionssystemen für den Computer auch oft vor – dass Ereignisse in anderer als einer textuellen Form, etwa mit Hilfe eines Bildes, einer Graphik, oder einem Ausschnitt aus einer Ton- oder Video-Datei beschrieben werden. Es kann nicht das Ziel eines Transkriptionssystems sein, Mechanismen für die *Kodierung* solcher Beschreibungen zur Verfügung zu stellen. Statt dessen kann es aber für die allermeisten Zwecke bereits genügen, im System eine Möglichkeit vorzusehen, auf solche Daten zu *verweisen*.[39] Ereignisse, die nicht mit Hilfe von Zeichenketten, sondern mit Hilfe von (Datei)verweisen beschrieben werden, werden in eine Kategorie des Typs ‚L' (für Link) eingeordnet.

Es ist keineswegs sicher, dass diese vier Typen die möglichen Ereigniskategorien bereits ausreichend charakterisieren, um alle denkbaren generischen Verarbeitungsschritte für Diskurstranskriptionen vornehmen zu können. Für die bisher implementierten EXMARaLDA-Komponenten haben sie sich aber bereits als hilfreich (und ausreichend) erweisen, und die Architektur des Systems schließt eine Erweiterung oder Verfeinerung der Typisierung nicht aus.

[39] Ein solcher Verweis erfolgt dann seinerseits wieder über eine Zeichenkette, und es existieren sogar Mechanismen (**U**niform **R**esource **I**dentifiers), diese Zeichenketten eindeutig interpretierbar zu machen.

Teil B: Technische Umsetzung

1. Geeignete Technologien

1.1. UNICODE

Die grundlegende Einheit einer Transkription ist – unabhängig von jeglichen weitergehenden strukturellen Überlegungen – Text. Wie oben kurz angesprochen, kann eine Transkription zwar durchaus nicht-textuelle Bestandteile wie Bilder, Ton oder Video enthalten, in erster Linie kann und muss eine Transkription aber als eine wie auch immer geartete Struktur von Zeichenketten betrachtet werden. Der Auswahl einer geeigneten Technologie zur Kodierung von Zeichenketten kommt daher eine essentielle Bedeutung zu.
Der allgemein akzeptierte und verwendete Standard zur Textkodierung war bis vor wenigen Jahren der sogenannte ASCII-Standard, der in seiner ursprünglichen Version (7-bit-ASCII) eine standardisierte Kodierung für 128 Zeichen (Groß- und Kleinbuchstaben des lateinischen Alphabetes, arabische Ziffern, die gängigsten Interpunktionszeichen und einige zusätzliche Zeichen) vorsah. Zusätzlich dazu wurden *mehrere* standardisierte Erweiterungen (8-bit-ASCII) auf 256 Zeichen eingeführt, um z.B. zusätzliche auf dem lateinischen Alphabet basierende Sonderzeichen (wie die deutschen Umlaute oder akzentuierte Buchstaben) oder nicht-lateinische Alphabete (wie z.B. das kyrillische oder das griechische Alphabet) kodieren zu können. Die Beschränkung auf 256 Zeichen stellte sich aber, vor allem im Kontext der zunehmenden Vernetzung, als zu stark heraus – beispielsweise existierte kein Standard zur Kodierung von Texten, die *sowohl* Zeichen aus dem lateinischen, *als auch* aus dem kyrillischen und griechischen Alphabet enthalten[40]. Textdaten waren durch diesen Umstand sprachabhängig.
Der seit wenigen Jahren vorhandene und von zunehmend mehr verschiedenen Betriebssystemen und Anwendungen unterstütze UNICODE-Standard kodiert darum 65536 Zeichen, d.h. er ermöglicht eine standardisierte Kodierung für den weitaus größten Teil aller Schriftsysteme. Für ein Transkriptionssystem ist dabei besonders interessant, dass UNICODE einerseits auch linguistisch motivierte Zeichensätze (insbesondere das IPA) enthält und andererseits ausdrücklich Bereiche für benutzer-definierte Zeichenkodierungen vorsieht. UNICODE wird von den im folgenden beschriebenen Technologien XML und JAVA voll unterstützt und kann deshalb als geeignete Technologie für eine weitestgehend sprachunabhängige Kodierung von Text, wie sie für ein Transkriptionssystem, das hinsichtlich der zur Transkription benutzten Sprachen und Zeichensysteme keinerlei Annahmen macht, zweifelsohne benötigt wird, angesehen werden.

1.2. XML

UNICODE standardisiert die Art und Weise, in der *einzelne Zeichen* auf dem Computer kodiert werden. Wie im ersten Kapitel dargestellt, sind Diskurstranskriptionen aber mehr als nur Zeichensequenzen – sie sind *strukturierte* Dokumente, d.h. sie organisieren Zeichen zu größeren, zueinander in Beziehung stehenden Einheiten. XML ist eine Sprache, die es erlaubt, sol-

[40] Für nicht-alphabetische Schriftsysteme, insbesondere die sogenannten CJK (Chinese-Japanese-Korean)-Systeme war die Zahl von 256 Zeichen sogar von vorneherein nicht ausreichend.

che Strukturen softwareunabhängig zu beschreiben. Im Gegensatz zu Auszeichnungssprachen wie HTML sind dabei in XML die Strukturelemente nicht festgelegt, sondern können anwendungsspezifisch definiert werden. Es würde den Rahmen dieses Papiers sprengen, die Möglichkeiten und Grenzen eines Einsatzes von XML zur Kodierung von Diskurstranskriptionen im einzelnen darzustellen. Folgende Punkte spielen dabei aber in jedem Falle eine wichtige Rolle:

- XML hat derzeit den Status einer W3C-Empfehlung. Es ist zu erwarten, dass es in der nächsten Zeit zu einem offiziellen (ISO-)Standard erklärt werden wird. Durch diesen Umstand ist nicht nur eine lange Lebensdauer von in XML kodierten Daten garantiert, es ist auch zu erwarten, dass langfristig und im großen Rahmen XML-Werkzeuge und auf XML aufbauende Technologien entwickelt werden.[41]
- viele (wahrscheinlich der weitaus größte Teil) der in der Entwicklung befindlichen Standards zur Repräsentation linguistischer Daten benutzen XML in irgendeiner Form.
- ein möglicher Einwand gegen die Benutzung von XML zur Kodierung von Diskurstranskriptionen ist seine Beschränkung auf streng hierarchische Strukturen (Baumstrukturen / kontextfreie Sprachen). Diese Beschränkung existiert aber nur vordergründig, d.h. sie bezieht sich auf die Grundstruktur von XML. Sie kann durch den Einsatz zusätzlicher Möglichkeiten, die teilweise direkt in XML integriert sind (IDs und IDREFs) und teilweise Gegenstand von *XML-related standards* (XLink, XPointer) sind, problemlos umgangen werden[42].

Für eine detailliertere Diskussion dieser Aspekte sei z.B. auf Bird/Liberman(2001), Dybkjaer et al. (1998a, 1998b), Dybkjaer (2000), Ide (2000), Isard (2001), Muhr (2000), Rehm/Lobin (2001) und Wohlberg (1999) verwiesen.

1.3. JAVA

Die Verwendung von XML zur Datenkodierung macht das System zur Diskurstranskription und –annotation weitestgehend softwareunabhängig und damit auch bis zu einem gewissen Grad betriebssystemunabhängig. Die Möglichkeit eines reinen Datenaustausches zwischen verschiedenen Betriebssystemen ist durch XML zum Beispiel bereits sichergestellt, und grundlegende Arbeitsschritte wie Such- und Ersetzvorgänge können mit vorhandener Software bewerkstelligt werden, die für verschiedene Betriebssysteme existiert. Trotzdem sind zu einer angemessenen Erstellung und Bearbeitung einer Diskurstranskription auch spezielle Software-Werkzeuge nötig, und diese müssen eigens für diese Zwecke erstellt werden. Bei vielen existierenden Systemen sind solche Werkzeuge auf bestimmte Betriebssysteme beschränkt[43]. Auch wenn die Daten dieser Systeme auf verschiedenen Betriebssystemen verwendbar wären (was ebenfalls nicht unbedingt der Fall ist), scheitert ihre Verwendung daher oft daran, dass die Werkzeuge für ihre Verarbeitung betriebssystemabhängig sind.
Als geeignete Programmiersprache für eine betriebssystemunabhängige Programmierung von Werkzeugen bietet sich JAVA an, denn

[41] Während also UNICODE die Kodierung auf Zeichenebene standardisiert, ist XML ein vielversprechender Kandidat für einen Standard zur Kodierung strukturierter Daten - „What unicode is doing for language support, XML is doing for data exchange.“ (http://www.filemaker.com/xml/overview.html)
[42] Vgl. dazu vor allem Wohlberg (1999) und Dybkjaer (2000).
[43] Z.B. MAC OS 9.x und früher für den syncWriter (Rehbein et al. (1993)), DOS für HIAT-DOS (Ehlich (1992)), MAC OS 9.x und früher / Windows für CLAN (MacWhinney(1995)) etc.

- JAVA-Anwendungen werden nicht direkt vom Betriebssystem, sondern von einer zwischengeschalteten „Virtuellen Maschine“ interpretiert. Eine solche virtuelle Maschine exisitiert für alle gängigen Betriebssysteme (Windows, MAC OS, Linux, Unix), wodurch eine einmal programmierte JAVA-Anwendung auf allen diesen Systemen lauffähig ist.[44]
- JAVA unterstützt die oben beschriebenen Technologien XML und UNICODE
- JAVA ist eine moderne, objektorientierte Programmiersprache und bietet damit eine Möglichkeit, Teile vorhandener Anwendungen sinnvoll wiederzuverwerten oder als Bibliotheken anderen Entwicklern zugänglich zu machen (und zwar wiederum unabhängig vom Betriebssystem). Damit werden Prozesse wie Erweiterung und Adaption vorhandener Software einfacher möglich.

1.4. Weitere Aspekte der Implementierung

Die Entscheidung, bei der Implementierung des Transkriptionssystems auf die Technologien UNICODE und XML zur Kodierung der Daten und JAVA zur Programmierung der Werkzeuge zurückzugreifen, ist zwar einerseits, wie oben dargelegt, nicht willkürlich, sie ist andererseits aber auch nicht zwingend. Während es nämlich zwar wahrscheinlich bei der Zeichenkodierung keine ernsthafte Alternative zu UNICODE gibt, ist XML keineswegs der einzige geeignete Standard zur Kodierung strukturierter Daten. Die offensichtlichsten möglichen Alternativen wären SQL oder SGML – vergleiche dazu vor allem Milde (1999), aber auch Knowles (1995). Die Implementierung der Werkzeuge in JAVA hat zwar den Vorteil der Betriebssystemunabhängigkeit, diese ist aber einerseits auch bei einigen anderen Programmiersprachen gewährleistet (z.B. tcl/tk oder perl), andererseits kann es sogar gute Gründe geben, JAVA als ungeeignet für die Programmierung bestimmter Tools anzusehen: z.B. lassen sich gewisse Aufgaben effizienter mit einfacheren Skript-Sprachen wie perl oder awk bewerkstelligen, andere Aufgaben mögen aufgrund ihres Umfangs auf „schnellere“ Sprachen als JAVA (wie z.B. C++) angewiesen sein. Die Architektur des Systems schließt aber die Verwendung solcher alternativer Technologien auch gar nicht aus. Vielmehr ist eine bisher unausgesprochene Idee hinter EXMARaLDA die Möglichkeit, das System von verschiedenen Entwicklern für spezifische Aufgaben erweitern zu können. Um eine solche verteilt Entwicklung zu ermöglichen, wurden und werden deshalb einige hierfür notwendige Prinzipien beachtet:

- soweit möglich werden Standards verwendet. Dies wird vor allem im Einsatz von XML und UNICODE widergespiegelt, aber auch z.B. in der Tatsache, dass die Darstellungskodierung in den weitverbreiteten und standardisierten Formaten HTML und RTF erfolgt. Der Umstand, dass EXMARaLDA Diskurstranskriptionen auf eine Art und Weise beschreibt, die zum Formalismus der Annotationsgraphen (Bird/Liberman (2001)) konform ist, könnte sich als eine weitere Nutzung eines Standards erweisen, wenn dieser Formalismus seine derzeitige Popularität beibehält oder sogar ausweitet (siehe hierzu vor allem [TALKBANK])
- die Architektur des Systems ist weitestgehend modular. Beispielsweise wird die Beschreibung der Datenstruktur dadurch modularisiert, dass zusätzlich zur umfassenden segmentierten Transkription auch zwei Teilmengen einer solchen – die Basis- und die Listen-Transkription – definiert werden. Je nach angestrebter Verwendung kann es für einen Entwickler ausreichend sein, sich auf eine dieser (einfacheren) Teilmengen zu

[44] Das ist zumindest die Idee hinter JAVA. Sie ist zur Zeit noch nicht umfassend umgesetzt: Mac OS – Betriebssysteme stellen (noch) keine JAVA-Maschinen für sämtliche JAVA-Versionen zur Verfügung.

beschränken. Beispielsweise kann sich ein Import-/Exportfilter für die Konvertierung zwischen EXMARaLDA-Daten und CHAT-konformen Transkriptionen wahrscheinlich auf die Teilmenge der Listen-Transkriptionen beschränken, während ein Partitur-Editor lediglich auf Basis-Transkriptionen operieren kann. JAVA als objekt-orientierte Programmiersprache erzwingt weiterhin bereits einen gewissen Grad an Modularisierung bei der Implementierung der Werkzeuge. Auch hier wird jedoch zusätzlich darauf geachtet, dass einzelne Software-Komponenten unabhängig voneinander wieder- und weiterzuverwenden sind.

- schließlich wird allgemein angestrebt, die Entwicklung des Systems möglichst offen vorzunehmen. Vergleiche Rehm/Lobin (2001) für eine Diskussion der in diesem Zusammenhang wichtigen Schlagworte „Open Source“ und „Open Information“

2. EXMARaLDA Version 1.0

Ausgehend von den oben dargestellten Überlegungen wurden erste Komponenten eines Systems zur Diskurstranskription und –annotation auf dem Computer implementiert, dessen Name EXMARaLDA (EXtensible MARkup Language for Discourse Annotation) den Umstand reflektieren soll, dass die zentrale Komponente des Systems eine XML-basierte Sprache zur (inhaltsbasierten) Kodierung von Transkriptionsdaten ist.

In der aktuellen Version besteht EXMARaLDA aus:

1. Drei Dokumenttypdefinitionen (DTD), die eine Syntax für die XML-Kodierung von Diskurstranskriptionen festlegen.

2. Mehreren JAVA-Routinen zum Lesen, Schreiben und Bearbeiten der XML-kodierten Transkriptionsdaten

3. Mehreren JAVA-Routinen für die Berechnung von verschiedenen HTML- oder RTF-kodierten Darstellungen des XML-kodierten Inhaltes

4. Einem ebenfalls in JAVA programmierten Prototyp eines Editors zum Anfertigen von Basis-Transkriptionen in einer Partitur-Oberfläche

5. Einer JAVA-Routine zum Import von EXMARaLDA-Transkriptionen aus Textdateien, in denen eine Diskurstranskription nach den Prinzipien des vertikalen Darstellungstyps kodiert ist

Es sollen an dieser Stelle keine architektonischen Details des Systems beschrieben werden[45], sondern lediglich festgehalten werden, wie die derzeitige Implementierung von EXMARaLDA die in im ersten Teil dieses Papiers gemachten Überlegungen umsetzt:

zu 1.: In den Dokumenttypdefinitionen wird gemäß der in Abschnitt 1 getroffenen Unterscheidung die *inhaltsseitige Kodierung* von Diskurstranskriptionen festgelegt, genauer:

[45] Dies wird in einer eigenen System-Dokumentation erfolgen, die über die Website des Sonderforschungsbereiches „Mehrsprachigkeit“ (http://www.rrz.uni-hamburg.de/SFB538/) zugänglich gemacht werden wird

- definiert die DTD „`basic-transcription.dtd`“ die Syntax einer XML-Kodierung der in Abschnitt A.2 beschriebenen *Basis-Transkription*

- definiert die DTD „`segmented-transcription.dtd`“ die Syntax einer XML-Kodierung der in Abschnitt A.4 beschriebenen *segmentierten Transkription*

- definiert die DTD „`list-transcription.dtd`“ die Syntax einer XML-Kodierung der ebenfalls in Abschnitt A.4 beschriebenen *Listen-Transkription*

Wie in Abschnitt A.4 angedeutet, können die durch diese DTDs beschriebenen Transkriptionen als Teilmengen voneinander betrachtet werden, es gilt:

`basic-transcription` $\subset$ `list-transcription` $\subset$ `segmented-transcription`

Basis-Transkriptionen sind grundsätzlich in Partitur- oder Spaltendarstellung darstellbar, Listen-Transkriptionen zusätzlich auch noch in vertikaler Darstellung. Eine segmentierte Transkription ist hingegen u.U. mit den herkömmlichen Darstellungsmethoden gar nicht vollständig darstellbar. Sie enthält jedoch die maximale Information und sollte somit die Grundlage für eine Archivierung, eine Weiterverarbeitung (z.B. Annotation) oder eine Auswertung sein.

zu 2.: zu den JAVA-Routinen zur Bearbeitung von EXMARaLDA-Daten gehören unter anderem

- Routinen zur automatischen Überführung einer Basis- oder Listen-Transkription in eine segmentierte Transkription und (soweit möglich) umgekehrt

- Routinen zur automatischen Segmentierung von Basis- oder Listen-Transkriptionen, d.h. Routinen, die gemäß den in Abschnitt A.4 dargelegten Überlegungen aus der zeitlichen Struktur eines Diskurses automatisch eine sprachliche Struktur berechnen.

Bei diesen Routinen wird jeweils von der in Abschnitt A.5 beschriebenen Typisierung von Ereigniskategorien Gebrauch gemacht.

zu 3.: in Abschnitt A.3 wurde dargelegt, wie aus der inhaltsseitigen Repräsentation von Transkriptionsdaten darstellungsseitige Repräsentationen der verschiedenen Darstellungstypen berechnet werden können. Die diesbezüglichen JAVA-Routinen in EXMARaLDA setzen diese Überlegungen um, genauer:

- kann aus der in einer Listen-Transkription kodierten Information eine vertikale Darstellung berechnet werden und diese in HTML oder RTF kodiert werden

- kann aus der in einer Basis-Transkription kodierten Information eine Partitur-Darstellung berechnet werden und diese ebenfalls in HTML oder RTF kodiert werden. Die hierfür notwendige Technik zum Umbrechen von Partiturflächen auf eine bestimmte Seitenbreite ist dabei bereits integriert.

Weitere JAVA-Routinen, insbesondere zur Berechnung einer Spalten-Darstellung, können und werden in späteren Versionen hinzugefügt werden.

zu 4 und 5: Bei der Implementierung des Editor-Prototyps und des Import-Filters wurde von der oben angesprochenen Modularisierung der Datenstruktur Gebrauch gemacht. Der Editor dient zunächst nur zum Erstellen von Basis-Transkriptionen, die dann aber mit den entsprechenden Werkzeugen in eine segmentierte Transkription überführt werden können. Der Import-Filter operiert hingegen auf Listen-Transkriptionen, die ebenfalls nach einer Überführung in eine segmentierte Transkritpion weiterbearbeitet werden können.

Durch seine Datenzentriertheit ist EXMARaLDA auf eine flexible Erweiterung und Weiterentwicklung angelegt. Zu den ins Auge gefassten Erweiterungen des Systems gehören vor allem die Anbindung an eine oder mehrere Datenbanken sowie die Entwicklung benutzerfreundlicher Tools zum Anfertigen von Transkriptionen und Hinzufügen von Annotationen.

Literatur

Bird, Steven / Liberman, Mark (2001): *A formal framework for linguistic annotation.* In: Speech Communication 33 (1,2), pp. 23-60.

Bloom, Lois (1993): *Transcription and Coding for Child Language Research: The Parts are More than the Whole.* In: Edwards / Lampert (1993), pp. 149-168.

Burnard, Lou (1995): *The Text Encoding Initiative: an overview.* In: Leech et al. (1995), pp. 69-81.

Crysman, Berthold (1995): *LAPSUS: A utility for the transcription of empirical data in language acquisition research.* Unveröffentlichtes Manuskript, Hamburg.

Dybkjær, Laila (2000): *Final Report.* MATE Deliverable D6.2. [http://mate.nis.sdu.dk]

Dybkjær, Laila et al. (1998a): *Supported Coding Schemes.* MATE Deliverable D1.1. [http://mate.nis.sdu.dk/]

Dybkjær, Laila et al. (1998b): *The MATE Markup Framework.* MATE Deliverable D1.2. [http://mate.nis.sdu.dk]

Edwards, Jane / Lampert, Martin (ed.) (1993): *Talking Data – Transcription and Coding in Discourse Research.* Hillsdale.

Edwards, Jane (1992): *Computer methods in child language research: four principles for the use of archived data.* In: Journal of Child Language 19, pp. 435-458.

Edwards Jane (1993): *Principles and Contrasting Systems of Discourse Transcription.* In: Edwards / Lampert (1993), pp. 3-31.

Ehlich, Konrad (1992): *Computergestütztes Transkribieren - das Verfahren HIAT-DOS.* In: Richter, Günther (Hrsg.) Methodische Grundfragen der Erforschung gesprochener Sprache, Frankfurt a.M.: P. Lang, 47-59

Ehlich, Konrad (1993): *HIAT - a Transcription System for Discourse Data.* In: Edwards / Lampert (1992), pp. 123-148.

Ehlich, Konrad / Rehbein, Jochen (1976): *Halbinterpretative Arbeitstranskriptionen (HIAT).* In: Linguistische Berichte 45, pp. 21-41.

Ide, Nancy (2000): *The XML Framework and Its Implications for the Development of Natural Language Processing Tools.* In: Proceedings of the COLING Workshop on Using Toolsets and Architectures to Build NLP Systems, Luxembourg, 5 August 2000.

Isard, Amy (2001): *An XML architecture for the HCRC Map Task Corpus.* In: Kühnlein, Peter / Rieser, Hannes / Zeevat, Hank (Hrsg.): BI-DIALOG 2001.

Johansson, Stig (1995): *The approach of the Text Encoding Initiative to the encoding of spoken discourse.* In: Leech et al. (1995), pp. 82-98.

Knowles, Gerry (1995): *Converting a corpus into a relational database: SEC becomes MARSEC.* In: Leech et al. (1995), pp. 208-219.

Leech, Geoffrey / Myers, Greg / Thomas, Jenny (ed.) (1995): *Spoken English on Computer: Transcription, Markup and Application.* Harlow: Longman.

Lobin, Henning (Hrsg.) (1999a): *Text im digitalen Medium. Linguistische Aspekte von Textdesign, Texttechnologie und Hypertext Engineering.* Wiesbaden: Westdeutscher Verlag.

Lobin, Henning (1999b): *Intelligente Dokumente. Linguistische Repräsentation komplexer Inhalte für die hypermediale Wissensvermittlung.* In: Lobin (1999a), pp. 155-178.

MacWhinney, Brian (1995): *CHAT Manual.* [http://childes.psy.cmu.edu/]

MacWhinney, Brian / Snow, Catherine (1992): *The wheat and the chaff: or four confusions regarding CHILDES.* In: Journal of Child Language 19, pp. 459-471.

Milde, Jan-Torsten (1999): *Effizientes Document Engineering sprachlicher Daten.* In: Lobin (1999a), pp. 197-220.

Muhr, Thomas (2000): *Increasing the Reusability of Qualitative Data with XML.* In: Forum Qualitative Sozialforschung 1(3). http://qualitative-research.net/fqs/fqs-eng.htm.

Ochs, Elinor (1979): *Transcription as theory.* In: Ochs, Elinor / Schieffelin, Bambi (eds.) (1979): Developmental pragmatics. New York: Academic.

Rehbein, Jochen / Grießhaber, Wilhelm / Löning, Petra / Hartung, Marion / Bührig, Kristin (1993): *Manual für das computergestützte Transkribieren mit dem Programm syncWRITER nach dem Verfahren der Halbinterpretativen Arbeitstranskriptionen (HIAT).* Hamburg.

Rehm, Georg / Lobin, Henning (2001): *From Open Source to Open Information: Collaborative Methods in Creating XML-based Markup Languages.* Electronic Publishing 2000, August 17th-19th, Kaliningrad/Svetlogorsk: Kaliningrad State University, Russia (International Federation for Information Processing and International Council for Computer Communication).

Selting, Margret et al. (1998): *Gesprächsanalytisches Transkriptionssystem (GAT).* In: Linguistische Berichte (173), pp. 91-122.

Sinclair, John (1995): *From Theory to Practice.* In: Leech et al. (1995), pp. 99-109.

Wohlberg, Tim (1999): *Hypertables – Entwicklung einer Strukturbeschreibungssprache für Tabellen in XML.* Diplomarbeit am Fachbereich Informatik der Universität Hamburg.

[TALKBANK]: http://www.talkbank.org